Defence Research and Development Canada
Recherche et développement pour la défense Canada

DRDC | RDDC

Detailed maintenance planning for military systems with random lead times and cannibalization

R. Zhang
NSERC Visiting Fellow

A. Ghanmi
DRDC – Centre for Operational Research and Analysis

Defence Research and Development Canada

Scientific Report
DRDC-RDDC-2014-R165
December 2014

Canada

Detailed maintenance planning for military systems with random lead times and cannibalization

R. Zhang
NSERC Visiting Fellow

A. Ghanmi
DRDC – Centre for Operational Research and Analysis

Defence Research and Development Canada
Scientific Report
DRDC-RDDC-2014-R165
December 2014

Abstract

Detailed maintenance planning under uncertainty is one of the most important topics in military research and practice. As one of the fastest ways to recover failed weapon systems, cannibalization operations are commonly applied by maintenance personnel. Due to additional complexities introduced by these operations, detailed maintenance decision making with cannibalization was rarely studied in the literature. This report proposed an analytic model for making repair decisions in a multi-stage uncertain environment at the operational level, where cannibalization operations are allowed and repair lead times are random. The study addresses the problem of maintenance planning for military systems with random lead times and independent failures. The objective of the problem is to maximize fleet reliabilities under operating costs constraints. A complementary problem that minimizes total operating costs under fleet reliabilities constraints was also examined. A polynomial algorithm was proposed to solve the minimization problem and determine optimal decision strategies. This algorithm could be used as a subroutine in a binary-search algorithm to solve the maximization problem. The obtained solutions were proved to be controllable in such a way that solutions with designated approximation ratios were achievable by running the algorithm in predictable run times.

Significance for defence and security

Under peaceful conditions, operating mangers in the Canadian Armed Forces (CAF) are encouraged to *dynamically find out best decisions for maintenance systems*, where the word of "best" refers to the balance between operating costs and fleet reliabilities. As one of the fastest operations to recover a weapon system, especially for critical parts, cannibalization is widely used by operating managers. For example, according to the Canadian Army Divestment Plan, the CAF National Procurement reductions will constrain the CAF to prematurely divest up to half of its Heavy Logistics Vehicle Wheeled, Light Support Vehicles Wheeled, and BandVangn to maintain operational readiness to meet CAF missions. This report developed an optimization model to find the best maintenance decisions with cannibalization and presented solution approaches to the model. Operating managers would use the model to make repair decisions at the operational level over a finite number of time periods.

Résumé

La planification de l'entretien détaillé en contexte d'incertitude est l'un des plus importants sujets dans les domaines de la recherche et de la pratique militaires. L'une des pratiques les plus couramment utilisées par le personnel de maintenance pour réparer des systèmes d'arme défectueux est la cannibalisation. En raison des difficultés que comporte ce procédé, le recours à la cannibalisation comme pratique d'entretien détaillé a rarement fait l'objet d'études. Dans le présent rapport, on propose un modèle analytique pour la prise de décisions opérationnelles concernant la réparation d'équipement en plusieurs étapes et en contexte d'incertitude lorsque la cannibalisation est autorisée et les délais de réparation sont variables. L'étude aborde la question de la planification de l'entretien des systèmes militaires avec des délais aléatoires et des défaillances indépendantes. L'objectif est de maximiser la fiabilité des flottes malgré les contraintes liées aux coûts d'exploitation. On s'est également penché sur le problème complémentaire de la minimisation des coûts d'exploitation avec des contraintes sur le plan de la fiabilité. Pour résoudre ce problème et trouver des stratégies pour la prise de décisions optimales, on a proposé un algorithme polynomial. Cet algorithme pourrait aussi être utilisé comme sous-programme dans un algorithme de recherche binaire afin de résoudre le problème de la maximisation de la fiabilité. Les résultats obtenus se sont avérés contrôlables, de sorte que les solutions ayant un rapport d'approximation étaient réalisables en exécutant l'algorithme pendant une durée prévisible.

Importance pour la défense et la sécurité

En contexte de paix, on encourage les gestionnaires de l'exploitation des Forces armées canadiennes (FAC) à prendre de façon dynamique les meilleures décisions possible relativement aux systèmes d'entretien. Prendre les meilleures décisions possible signifie ici de trouver un équilibre entre les coûts d'exploitation et la fiabilité de la flotte. La cannibalisation est très souvent utilisée par les gestionnaires de l'exploitation, car c'est l'une des méthodes les plus rapides pour remettre un système d'arme à neuf, en particulier lorsqu'il s'agit du remplacement de pièces essentielles. Par exemple, selon le Plan de dessaisissement de l'Armée canadienne, les réductions dans l'approvisionnement national contraindront les FAC à se départir prématurément de près de la moitié de ses véhicules logistiques lourds à roues, de ses véhicules de soutien légers à roues et de ses BandVagn pour maintenir leur préparation opérationnelle en vue d'accomplir leurs missions. Dans le présent rapport, on présente un modèle d'optimisation appuyant la prise des meilleures décisions possible en ce qui a trait à l'entretien au moyen de la cannibalisation. On présente également les démarches employées pour trouver des solutions à l'aide du modèle. Les gestionnaires de l'exploitation feront usage de ce modèle pour prendre des décisions opérationnelles concernant la réparation d'équipement.

Acknowledgements

The authors would like to thank Dr. Gregory H. van Bavel for his advice and discussions on possible applications of this research to the CAF maintenance networks.

This page intentionally left blank.

Table of contents

List of figures

List of tables

1 Introduction

1.1 Background

In a military maintenance network, when a weapon system or prime equipment (PE) is malfunctioning, one or more responsible parts are identified and replaced. The failed parts will be separated from the PE and sent to a central depot for repair. The left holes are supposed to be filled with functioning parts. If spare parts are not available, cannibalizing functioning parts from other failed PEs is sometimes applied in practice. Figures 1 and 2 describe a cannibalization example. The example has a fleet with three PEs: PE-1, PE-2 and PE-3. Each PE is made up of three parts/LRUs (line replaceable units) in three different types: L-1, L-2 and L-3. In the figures, if an LRU is functioning, it is denoted by a solid-double-line rectangle; otherwise a dashed-double-line rectangle is used. Figure 1 has PE-1, PE-2 and PE-3 waiting for L-3, L-2 and L-1 replacement, respectively. The two arrows indicate that PE-1 and PE-3 can cannibalize L-3 and L-1 from PE-2, respectively. Using cannibalization operations, PE-1 and PE-3 can get back to work immediately, while PE-2 will be in an even worse situation, i.e., there are three LRU "holes" on PE-2. Figure 2 shows the status of the fleet after cannibalization. It is clearly observed that the number of functioning PEs increases from zero (Figure 1) to two (Figure 2). Certainly, the reliability level of the fleet is improved.

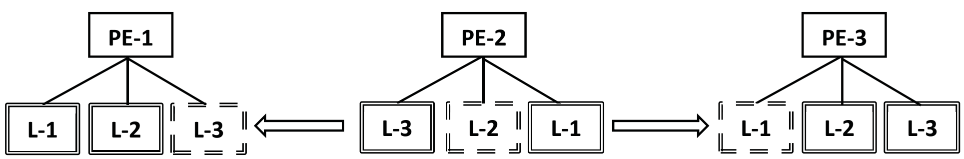


Figure 1: *PE-1 and PE-3 can cannibalize L-3 and L-1 from PE-2, respectively.*

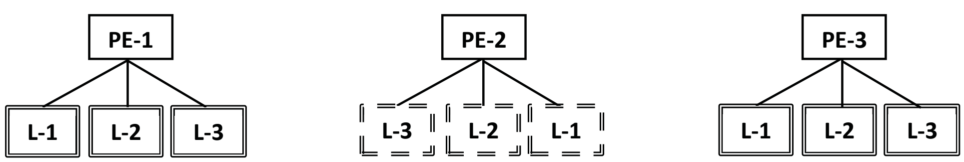


Figure 2: *PE-1 and PE-3 are recovered after cannibalization.*

This example demonstrates that cannibalization is a useful operation when there is no sufficient spare LRU; this operation is favored by maintenance personnel in practice. In a study by General Accounting Office of the United States (GAO U.S.) [1], Air Force and Navy units reported a total of about 850,000 cannibalizations on aircraft in fiscal years 1996-2000, including 376,000 cannibalizations by the Air Force and 468,000 by the Navy. For the aircraft B1-B of the Air Force, the cannibalization rate (the number of cannibalizations per 100 sorties) could be up to 85.4. For the aircraft F-14D of the Navy, the rate (the number of cannibalizations per 100 flight hours) could be up to 32.8. As such, cannibalization has been one of the most important topics in military research and practice since the 1970s.

By US DoD (the Department of Defense, United States) [2], there were 14,800 aircraft, 896 strategic missiles, 256 ships and 386,600 ground combat and tactical vehicles under the 2013 daily maintenance operations. The base of PE failure is huge such that a few number of malfunctioning PEs won't make substantial differences on actually failure rates. Based on this, a large number of literature assumed constant failure rates and therefore they were able to apply queue theory on maintenance and reliability study [3–7]. In the CAF reality, however, the number of PEs is usually very small (e.g., the Royal Canadian Navy operates four submarines, three destroyers and two replenishment vessels). Any failed PEs might cause substantial decreasing on the actual base of PE failures. This gives dynamically varied failure rates, i.e., the rate of PE failures in a time period depends on the number of functioning PEs at the beginning of the period. Using the theory of Markov decision process (MDP), Zhang [8] presented a multi-stage model, in which detailed cannibalization/repair decisions were made in each time period/stage to maximize total weighted number of functioning PEs over a finite number of decision periods. As a follow-up study, this report considers independent PE failures; that is, from the Probability Theory perspective, the behaviors of PE failures during each time period are assumed to be independent and identically distributed (IID) random variables.

1.2 Aim

Operating managers in CAF are required to follow a defined decision policy under wartime conditions. However, *under peaceful conditions they are encouraged to dynamically find out which decision is best*, where the word of "best" indicates the balance between "reasonable" operating costs and "acceptable" fleet reliabilities. The objective of this report is to develop a model to determine optimal decision strategies for maintenance planning of military systems with lead times and cannibalization.

1.3 Scope

There are two major reasons for undergoing cannibalization [1] – inefficient support system, and high readiness requirement and operational demands. This indicates that most cannibalization operations are driven by the mismatch between LRU demands and supplies. As an extreme case, LRU supply can be discontinued (i.e., no external supplies for LRUs). For example, one of the ongoing projects in CAF is to cannibalize its Eryx missile systems as it prepares for the eventual retirement of the weapon in 2016 [9]. This report models this particular case by assuming that the maintenance network does not include any third-party LRU supplies/demands; LRUs can come to (or leave) the network only when they are installed on some PEs and these host PEs come to the network as failed PEs (or leave it as functioning PEs).

In most situations, a maintenance network is comprised of one or more operating bases to hold all PEs and spare LRUs and a central depot to conduct all repair operations. Since the central depot is usually located in a separate site from the operating bases, base-depot transportation and depot-repair operations are out of control of operating managers. This report considers repair times and transportation costs as uncertain factors, i.e., repair lead times and transportation costs are not available until the times/periods come. Assuming that all base operations (i.e., LRU separation, installation and warehousing operations) are controlled by operating managers, this report focuses on repair decisions to answer the question: how many failed LRUs should be sent to the depot for repair at each time period.

Traditionally, maintenance planning problems of systems with IID-featured PE failures are studied using Queuing Theory [3, 4, 10]. However, this method cannot be used for detailed operational decision making. In this report, stochastic optimization approaches [11, 12] are used to formulate and solve the maintenance planning problem as a multi-stage decision making problem.

1.4 Structure

This report is organized as follows. Section 2 presents a literature review for cannibalization, a description of the maintenance system, and a formal definition of the problem for maximizing fleet reliabilities. In Section 3, in order to solve the maximization problem, a complementary problem that determines a set of multi-stage repair decisions for minimizing total operating costs with constraints on fleet reliabilities is constructed. A dynamic programming algorithm is then developed to solve the minimization problem. In Section 4, a binary-search algorithm is developed to provide approximation solutions to the maximization problem. Finally, conclusions and future research are presented in Section 5.

2 Preliminaries and MaxRho

2.1 Literature review

Manufactured products can fail due to different processes such as corrosion, wear and tear, and fatigue. Products such as domestic electronics and appliances are generally discarded and replaced upon failure because they are inexpensive. However, capital goods such as defense weapon systems (or PEs) are repaired because of their high replacement costs. Repair/maintenance decisions often involve the removal and replacement of the failed parts. Research on maintenance-decision-making problems have spawned several noteworthy papers in the open literature over the last five decades, starting with the textbook [13] (which fundamentals of maintenance and reliability research methods were established) and the paper [14] (important earlier developments in this research area were reviewed). Most research works before the middle of 1980's were reviewed in the survey papers [15–17]. Later, Valdez-Flores and Feldman [18] and Cho and Parlar [19] reviewed the research works for single and multiple items maintenance decision making, respectively. For more recent developments, the review papers [20], [21] and [22] surveyed studies on corrective, preventive and opportunistic maintenance policies, respectively.

In the military context, the main points of interest are the level of repair analysis (LORA) problems (i.e., the relocation decisions of repair facilities on a support network [23, 24]), the spare parts stocking (SPS) problems (i.e., the initial inventory decisions of spare parts on a support network [25, 26]) and the combined LORA-SPS problems on a support network [27]. In these problems, repair decisions were either studied at the strategic level using aggregated approaches (e.g., LORA or LORA-SPS), or even not considered at all (e.g., SPS). However, operating managers are under increased pressures to improve fleet reliabilities through their detailed (or daily) repair decisions. Thus, there is a need to study multi-stage repair decision making for military maintenance networks at the operational level.

Detailed operational planning in a multi-stage, uncertain environment is one of the most important and difficult topics in operations research. With respect to different uncertainty features, two major solution approaches are considered: (1) the stochastic dynamic programming (SDP) approach [28], in which a decision tree is first constructed and then the tree is solved backwards, is a common method to solve MDP or MDP-featured problems, and (2) the scenario tree based (STB) approach [29], in which a deterministic scenario tree formulation is first presented and then a dynamic programming algorithm is developed to find optimal solutions on the scenario tree. Since the maintenance systems studied in this report have IID-featured PE failures, the structure of scenario tree used in the formulation can be easily determined in advance. This report adopts the STB approach for later algorithm design and analysis. Note that the SDP approach was used in [8] for maintenance systems with MDP-featured PE failures.

2.2 System description

The report considers a single-base, single-depot maintenance network, which is described in Figure 3. In the network, all failed PEs and spare LRUs are installed in the operating base and all repair operations are conducted in the depot (assuming that the warehouse space in the base and the repair capacity in the depot are unlimited). Initially, there are N $(0 < N < \infty)$ failed PEs $(E_1, E_2, \cdots, E_N)$ installed in the base and each PE is made up of M $(1 < M < \infty)$ distinct LRUs $(L_1, L_2, \cdots, L_M)$. Note that among these M LRU positions there is at least one missing LRU hole or malfunctioning LRU for each failed PE.

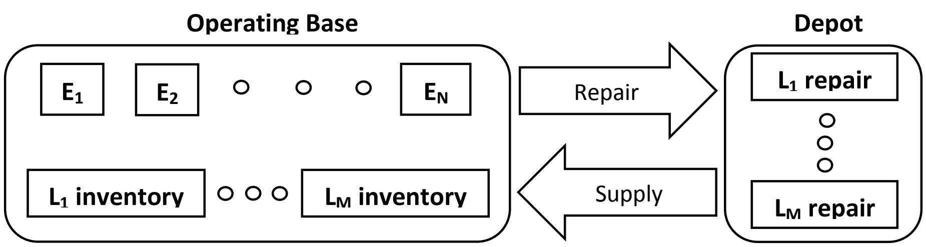


Figure 3: *The support network has one operating base and one repair depot.*

Let f_m and g_m be the number of functioning and malfunctioning L_m in the base, respectively, where $m = 1, 2, \cdots, M$. Let q_m be the number of due-in L_m, which are either under repair in the depot or in transshipment between the base and the depot. Since they are not located in the base, they are not included in either f_m or g_m. Thus, the total number of spare (or *individually existed*) L_m can be calculated as:

$$s_m = f_m + g_m + q_m - N, \quad m = 1, 2, \cdots, M. \tag{1}$$

The use of words "individually existed" is due to the observation that each PE should include a PE frame, on which LRUs are installed. When the extra LRUs are first introduced in the system, they are not attached to any PE frame. Assuming that there are no external supplies or demands for any individual LRUs, s_m for all $m = 1, 2, \cdots, M$ remain constant during the whole decision horizon.

A system is called *effective* if all failed PEs, whose L_m demands can be satisfied using either the stocked L_m or the functioning L_m on other failed PEs (via cannibalization), are recovered. Given that all in-base operations are well-controlled by operating managers, PE-recovery operations can be treated as zero-time and zero-cost operations.

Assuming that the initial system is effective, $N > 0$ implies that there is at least one failed LRU (say L_m) such that $f_m = 0$, i.e., $\min\{f_1, f_2, \cdots, f_M\} = 0$.

Let $[1, T+1)$, where $1 \leq T < \infty$, be the decision horizon including T mutually disjoint time periods. For instance, Figure 4 depicts the repair decisions and PE failures of period t (where $1 \leq t \leq T$), i.e., all repair decisions are assumed to be made at the beginning of period t (just after time t) and all PE failures are assumed to be realized at the end of period t (just before time $t+1$).

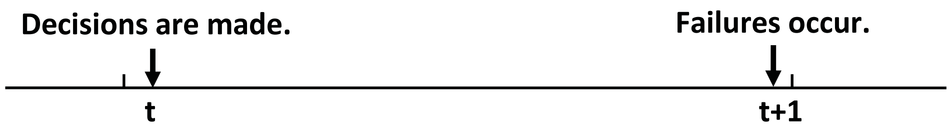


Figure 4: *The repair decisions and PE failures in period t.*

Let $\{\mathbf{f}^t, \mathbf{g}^t, \mathbf{q}^t, n^t\}$ denote the system status at time t, where $\mathbf{f}^t = (f_1^t, f_2^t, \cdots, f_M^t)$, $\mathbf{g}^t = (g_1^t, g_2^t, \cdots, g_M^t)$, $\mathbf{q}^t = (q_1^t, q_2^t, \cdots, q_M^t)$ and $n^t = f_m^t + g_m^t + q_m^t - s_m$ with $m = 1, 2, \cdots, M$. As a consequence of the effective-system assumption, the PEs, which are recovered during period t, are assumed to leave the system by the end of period t (i.e., just before time $t+1$ in the figure). Thus, at time $t+1$ only the number of un-covered/failed PEs is recorded, which is denoted by n^{t+1} with $n^{t+1} = f_m^{t+1} + g_m^{t+1} + q_m^{t+1} - s_m$, $\forall m$. In order to keep notation consistency, the initial status at time one can be re-denoted by the 1-superscripted notations, i.e., $f_m^1 = f_m$, $g_m^1 = g_m$, $q_m^1 = q_m$, and $n^1 = N$.

2.3 Failures and repairs

Assume that IID-featured PE failures follow the same Homogeneous Poison process (HPP) with mean $\lambda > 0$ in all periods. By HPP, the probability of PE failures with k failed PEs can be written as:

$$\frac{\lambda^k e^{-\lambda}}{k!}, \quad \text{where } k = 0, 1, \cdots. \tag{2}$$

Let $0 < \gamma < 1$ be the threshold probability such that PE failures with probabilities smaller than γ would be considered as zero-probability events. Considering a reasonable γ value (i.e., $\gamma \ll 1$ is a relatively small value compared to value 1), HPP-distributed PE failures have smaller probabilities as k (the number of failed PEs) gets larger. Let K be the largest integer value such that $\frac{\lambda^K e^{-\lambda}}{K!} \geq \gamma$. This implies that PE failures with more than K failed PEs are considered as zero-probability events. Using this assumption, the PE-failure probabilities described in Equation (2) can be modified as:

$$\text{Prob}(k) = \frac{\frac{\lambda^k e^{-\lambda}}{k!}}{\sum_{k'=0}^{K} \frac{\lambda^{k'} e^{-\lambda}}{k'!}}, \quad \text{where } k = 0, 1, \cdots, K. \tag{3}$$

It is assumed that there is one and only one responsible LRU for the PE failure with exactly one failed PE. Let $\beta_m \in [0,1]$ be the probability of such a PE failure caused by a failed L_m. Thus, $\sum_{m=1}^{M} \beta_m = 1$. Let $\mathbf{k} = (k_1, k_2, \cdots, k_M)$ be a vector denoting a PE failure with k failed PEs, where $k_m \geq 0$ is the number of failed PEs (whose responsible LRUs are L_m), and

$$k = \sum_{m=1}^{M} k_m. \tag{4}$$

is the total number of failed PEs. By letting $k_0 = 0$, the probability of such a PE failure can be written as:

$$\text{Prob}(k_1, k_2, \cdots, k_M : k) = \text{Prob}(k) \prod_{m=1}^{M} \binom{k - \sum_{m'=0}^{m-1} k_{m'}}{k_m} (\beta_m)^{k_m}. \tag{5}$$

It is easy to prove that Equation (5) is a probability mass function satisfying the following two fundamental conditions in Discrete Probability Theory:

$$0 \leq \text{Prob}(k_1, k_2, \cdots, k_M : k) \leq 1, \tag{6}$$

and

$$1 = \sum_{(k_1, k_2, \cdots, k_M) \in \mathcal{K}(k)} \sum_{k=0}^{K} \text{Prob}(k_1, k_2, \cdots, k_M : k), \tag{7}$$

where $\mathcal{K}(k) = \{(k_1, k_2, \cdots, k_M) | k = \sum_{m=1}^{M} k_m, k_m \geq 0, \forall m\}$ and $k = 0, 1, \cdots, K$.

Consider period t, let x_m^t be the number of failed L_m sent to the depot for repair. The zero-cost and zero-time assumption for all in-base operations implies that x_m^t (where $0 \leq x_m^t \leq g_m^t$ with $m = 1, 2, \cdots, M$) are the only decision variables for repair operations in period t. Let $\mathbf{k}^t = (k_1^t, k_2^t, \cdots, k_M^t)$ be the PE failure occurring in period t, where $k^t = \sum_{m=1}^{M} k_m^t$. Thus, in the base, the total number of malfunctioning L_m, which are the resources of L_m-repair operations in period $t+1$, can be calculated as:

$$\hat{g}_m^t = g_m^t - x_m^t + k_m^t. \tag{8}$$

Since repair and transshipment operations are not controlled by operating managers, random repair lead times are assumed. Let d_m^t be the lead time of L_m repair in period t. (Note that lead times are assumed to be *non-crossover*, i.e., $t_1 + d_m^{t_1} \le t_2 + d_m^{t_2}$ if $t_1 < t_2$.) Let $\mathcal{A}_m^t = \{\tau | \tau + d_m^\tau = t\}$ be the set including all periods, in which if a failed L_m is sent to the depot for repair it will be repaired and returned to the base in period t. Therefore, taking into account all functioning LRUs on the newly-failed PEs in period t, the total number of functioning L_m in the base is:

$$\hat{f}_m^t = f_m^t + k^t - k_m^t + \sum_{\tau \in \mathcal{A}_m^t} x_m^\tau. \tag{9}$$

Using the effective-system assumption, the number of recovered PEs would be:

$$\begin{aligned} \hat{n}^t &= \min\{n^t + k^t, \hat{f}_1^t, \hat{f}_2^t, \cdots, \hat{f}_M^t\} && (10) \\ &= \min\{n^t + k^t, \min_{m=1,2,\cdots,M}\{f_m^t + k^t - k_m^t + \sum_{\tau \in \mathcal{A}_m^t} x_m^\tau\}\}. && (11) \end{aligned}$$

As assumed, the recovered PEs are supposed to leave the system by the end of period t. Thus, the system is updated to $(\mathbf{f}^{t+1}, \mathbf{g}^{t+1}, \mathbf{q}^{t+1}, n^{t+1})$, where

$$\begin{aligned} f_m^{t+1} &= \hat{f}_m^t - \hat{n}^t = f_m^t + k^t - k_m^t + \sum_{\tau \in \mathcal{A}_m^t} x_m^\tau - \hat{n}^t, \quad \forall m && (12) \\ g_m^{t+1} &= \hat{g}_m^t = g_m^t - x_m^t + k_m^t, \quad \forall m && (13) \\ q_m^{t+1} &= q_m^t + x_m^t - \sum_{\tau \in \mathcal{A}_m^t} x_m^\tau, \quad \forall m && (14) \\ n^{t+1} &= n^t + k^t - \hat{n}^t. && (15) \end{aligned}$$

2.4 Objective and constraints

Since all in-based operations are well-controlled by operating managers, this report considers out-base costs only: LRU transshipment and repair costs. Let c^t be the total operating cost of period t, which is the sum of transportation (fixed and period-dependent) and repair (variable and LRU-dependent) costs. Let h^t be the transportation cost of period t if there is some malfunctioning LRUs sent to the depot for repair and let r_m be the per-LRU repair cost for the L_m-repair operations in period t. Thus, the total operating cost in period t can be written as:

$$c^t = h^t y^t + \sum_{m=1}^{M} r_m x_m^t, \tag{16}$$

where y^t is a binary decision variable such as $y^t = 1$ if there is at least one $x_m^t > 0$ for some $m \in \{1, 2, \cdots, M\}$ and $y^t = 0$ if $x_m^t = 0$ for all $m = 1, 2, \cdots, M$.

Let α (where $\alpha > 0$) be the interest gained on per unit unused fund for each period. Considering T periods, at time one the present value of total operating cost can be written as:

$$c = \sum_{t=1}^{T} \frac{c^t}{(1+\alpha)^{t-1}}. \tag{17}$$

Consider fleet reliabilities, which can be evaluated by recovering ratios. Let ρ^t be the recovering ratio of the first t periods, where $t = 1, 2, \cdots, T$. Thus,

$$\rho^t = \frac{\sum_{\tau=1}^{t+1} \hat{n}^\tau}{n^1 + \sum_{\tau=1}^{t+1} k^\tau}. \tag{18}$$

Adding a positive value of n^1 as part of the denominator is to avoid a possible zero denominator, i.e., ρ^t are well-defined for all $t = 1, 2, \cdots, T$.

Let B (where $0 < B < \infty$) be the total operating budget. The problem that maximizes fleet reliabilities with constraints on total operating costs (denoted by MaxRho) can be formulated as:

$$\max: \quad \rho \quad \text{s.t.:} \quad c \leq B \text{ and } \rho^t \geq \rho, \quad \forall t = 1, 2, \cdots, T, \tag{19}$$

where at the end of each time period systems are updated using the calculations described in Equations (12), (13), (14) and (15).

In order to solve MaxRho, a complementary problem is constructed to minimize total operating costs with constraints on fleet reliabilities. This problem is denoted by MinCost and it can be written as:

$$\min: \quad c \quad \text{s.t.:} \quad \rho^t \geq A, \quad \forall t = 1, 2, \cdots, T, \tag{20}$$

where $0 < A < \infty$ is the designated reliability level. Again, at the end of each time period systems are updated using the calculations described in Equations (12), (13), (14) and (15).

In the following, MinCost is first formulated on a scenario tree and solved using a dynamic programming algorithm. Then, MaxRho is solved approximately by solving a series of MinCost, whose designated reliability levels are dynamically updated.

3 MinCost and approaches

3.1 Scenario tree formulation

Using the commonly-adopted solution strategy (i.e., the STB approach), multi-stage stochastic optimization (MSO) problems are solved in two steps: (1) A MSO problem is represented as a scenario tree model (STM), which is the deterministic equivalence of the MSO problem; (2) A dynamic programming algorithm is designed to search for optimal solutions on the scenario tree and the found solutions are actually the solutions to the MSO problem. Next, a formal STM formulation for MinCost, denoted by MinCostSTM, is presented.

The T-stage MinCost model can be described using a $(T+1)$-level scenario tree, whose branches are constructed based on PE failures. Let $\mathcal{T}$ be the tree. Let $\mathcal{V}$ be the set of nodes on $\mathcal{T}$ and let $|\mathcal{V}| = \sum_{t=0}^{T+1} \Gamma^t$ be the number of nodes in $\mathcal{V}$, where $\Gamma = \sum_{k=0}^{K} |\mathcal{K}(k)|$ is the number of distinct PE failures during each time period (i.e., the number of branches on each *non-leaf* node, whose definition will be given later) and $|\mathcal{K}(k)|$ is the number of distinct PE failures with exactly k failed PEs ($k = 0, 1, \cdots, K$). In this report, $|\mathcal{V}|$ is used to denote the size of $\mathcal{T}$, which also is the size of MinCostSTM. Note that $|\mathcal{V}|$ is exponential on T (which is the number of time periods in MinCostSTM).

Recall that M (where $M > 0$) is the number of distinct LRU positions on each PE. It was assumed that among these M LRUs there is one and only one responsible LRU for the PE failure with exactly one failed PE. Therefore, by Combinatorial Number Theory [30–32], $|\mathcal{K}(k)|$ can be calculated using the following recursion formula:

$$|\mathcal{K}(k)| = \text{recPro}(k, M) = \sum_{\bar{M}=1}^{M} \text{recPro}(k - M, \bar{M}), \tag{21}$$

where the boundary condition is $\text{recPro}(0, \bar{M}) = 1$. Therefore, Γ can be re-written as:

$$\Gamma = \sum_{k=0}^{K} |\mathcal{K}(k)| = \sum_{k=0}^{K} \text{recPro}(k, M) = \sum_{k=0}^{K} \sum_{\bar{M}=1}^{M} \text{recPro}(k - M, \bar{M}). \tag{22}$$

Consider an example that each PE is made up of two distinct LRUs (i.e., $M = 2$) and the maximum of failed PEs in each time period is five (i.e., $K = 5$). Following the calculations in Equations (21) and (22) gives $\Gamma = 1 + 2 + 3 + 4 + 5 + 6 = 21$.

Let $l(j)$ be the level of node $j \in \mathcal{V}$. If $l(j) = T + 1$, then node j is called a leaf node. Reserving index 1 for the root node gives $l(1) = 1$. Let $a(j)$ be the direct ancestor node of node j and let $\mathcal{D}(j) = \{i | j = a(i), \forall i \in \mathcal{V}\}$ be the set of direct descendant nodes of node j. In particular, $a(1) = \emptyset$ and $\mathcal{D}(j) = \emptyset$ for all leaf nodes j. Let $\mathcal{T}(j)$ be the subtree rooted on node j and let $\mathcal{V}(j)$ be the set of nodes on $\mathcal{T}(j)$. This gives alternatives: $\mathcal{T}=\mathcal{T}(1)$ and $\mathcal{V}=\mathcal{V}(1)$.

Let $\mathcal{C}(j,l)=\{i|l(i)=l(j)+l,\forall i\in\mathcal{V}(j)\}$ be the set of nodes, which are in $\mathcal{V}(j)$ but with l levels higher than node j. In particular, $\mathcal{C}(j,1)=\mathcal{D}(j)$. There is no doubt that the leftover malfunctioning LRUs on node j can be repaired on any node in $\mathcal{C}(j,1)$ (or $\mathcal{D}(j)$). Let $\mathcal{L}(l)=\{i|l(i)=l,\forall i\in\mathcal{V}\}$ be the set of nodes on $\mathcal{T}$ with level l, where $l=1,2,\cdots,T+1$. If failed L_m are sent to the depot for repair on node j, then they will be repaired and arrive on all nodes in $\mathcal{V}(j)$ with level $l(j)+d_{(j,m)}$, i.e., on all nodes in $\mathcal{C}(j,d_{(j,m)})$. Note that $\mathcal{C}(j,d_{(j,m)})=\mathcal{L}(l(j)+d_{(j,m)})\cap\mathcal{V}(j)$.

Considering scenarios, let $\mathcal{P}(j)$ be the path from node 1 to node j including nodes 1 and j. If j is a leaf node, then $\mathcal{P}(j)$ *indeed* denotes a scenario. The number of scenarios on $\mathcal{T}$ is just the number of leaf nodes on $\mathcal{T}$ (which can be calculated by Γ^T). This is due to that $\mathcal{T}$ has $T+1$ levels and each non-leaf node has exactly Γ branches (assuming IID-featured PE failures). A scenario or path is a series of events and the probability of a scenario or path is represented by the probability of the node, with which the scenario or path ends up. For instance, let p_j be the probability of node $j\in\mathcal{V}$, then p_j is the probability of $\mathcal{P}(j)$. All reliability and cost calculations are based on either paths or scenarios.

To illustrate the above notations, a scenario tree example is given in Figure 5. Node i is the direct ancestor of node j, i.e., $i=a(j)$. Nodes j_{k_1}, j_{k_2} and j_{k_3} are the direct descendant nodes of node j, i.e., $\mathcal{D}(j)=\{j_{k_1},j_{k_2},j_{k_3}\}$. In the figure, $\mathcal{T}(j_{k_2})$ is a subtree rooted on node j_{k_2} and all nodes in $\mathcal{T}(j_{k_2})$ form $\mathcal{V}(j_{k_2})$. Path $\mathcal{P}(j)$, which includes node j and all ancestor nodes of j, is included as a common subpath by the nine scenarios presented in the figure. Obviously, this figure does not depict all nodes, branches or scenarios. As can be seen, node j_h is three levels higher than node j. This gives $\mathcal{C}(j,3)=\mathcal{V}(j)\cap\mathcal{L}(l(j)+3)=\mathcal{V}(j)\cap\mathcal{L}(l(j_h))=\{j_h,j_{h_1},j_{h_2},\cdots,j_{h_8}\}$.

Considering probabilities, since node 1 denotes the initial status, which is well-defined, the probability is $p_1=1$. Since all nodes in level 2 are branched from node 1,

$$\sum_{j\in\mathcal{D}(1)\ (\text{or } j\in\mathcal{L}(2) \text{ or } j\in\mathcal{C}(1,1))} p_j=p_1=1. \tag{23}$$

More generally, $\sum\limits_{i\in\mathcal{D}(j)} p_i=p_j$ is true for all non-leaf node j. Therefore, the sum of probabilities for all nodes on the same level is one, i.e., $\sum\limits_{j:l(j)=l} p_j=1$, and the sum of probabilities for all scenarios is one, i.e., $\sum\limits_{j:l(j)=T+1} p_j=1$. Next, MinCostSTM is formulated on a $(T+1)$-level scenario tree, which is similar to Figure 5.

In MinCostSTM, instead of using the notations introduced in Sections 2.2, 2.3 and 2.4, the following notations are used to denote the initial system on node one, i.e., $n_1=n^1$,

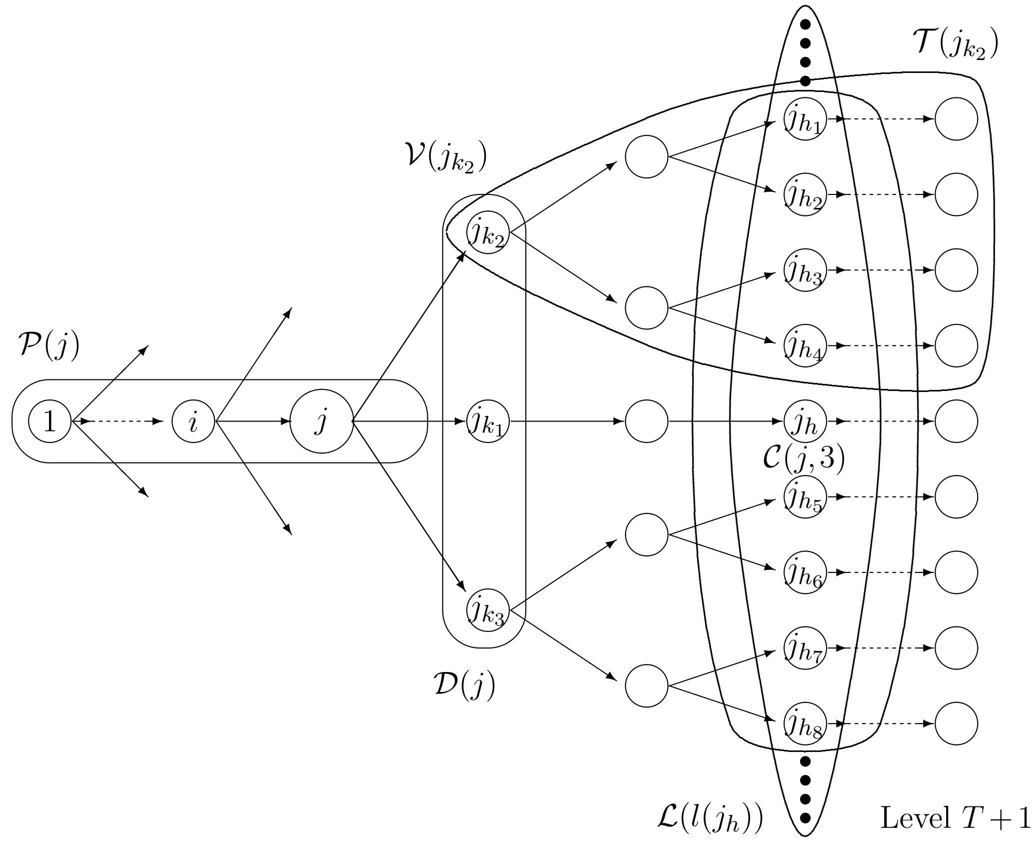


Figure 5: *Scenario tree notations:* $\mathcal{P}(j)$, $\mathcal{T}(j_{k_2})$, $\mathcal{V}(j_{k_2})$, $\mathcal{D}(j)$, $\mathcal{C}(j,3)$ *and* $\mathcal{L}(l(j_h))$.

$\mathbf{f}_1 = \mathbf{f}^1$, $\mathbf{g}_1 = \mathbf{g}^1$ and $\mathbf{q}_1 = \mathbf{q}^1$. Without loss of generality, $\mathbf{q}_1 = (0,0,\cdots,0)$ is assumed. Generally, let $\mathbf{D}_j = (D_{(j,1)}, D_{(j,2)}, \cdots, D_{(j,M)})$ be the cumulative LRU demands by node j, $\forall j \in \mathcal{V}$; that is when $D_{(j,m)}$ for all $m = 1,2,\cdots,M$ are satisfied, the fleet reliability constraint on node j is satisfied, i.e., $\rho_j \geq A$, where A is the required level for fleet reliabilities. For simplicity, "cumulative demand" is replaced with "demand" in the remaining part of the report.

In $\mathcal{T}$, PE failures are denoted by branches. For instance, in Figure 5, the branch connecting node j and j_{k_1} denotes the PE failures occurring in time period $[l(j), l(j_{k_1}))$. Since no branch heads into node 1, $\mathbf{D}_1$ is undefined and simply set as $\mathbf{D}_1 = (0,0,\cdots,0)$. In order to determine LRU demands, consider a scenario $\mathcal{P}(\bar{j})$. For any node $j \in \mathcal{P}(\bar{j})$ with $j > 1$, let $\bar{\mathbf{D}}_j = (\bar{D}_{(j,1)}, \bar{D}_{(j,2)}, \cdots, \bar{D}_{(j,M)})$ be the fundamental demand (F-demand) by node j. Let $\mathbf{k}_i = (k_{(i,1)}, k_{(i,2)}, \cdots, k_{(i,M)})$ be the PE failures denoted by the branches heading into node i for all $i \in \mathcal{P}(j) \setminus \{1\}$, where $k_{(i,m)}$ is the number of failed PEs due to malfunctioning L_m. In order to satisfy the fleet reliability constraints, the total number of recovered PEs by node j is at least:

$$\bar{n}(j) = \lceil A \times n(j) \rceil, \tag{24}$$

where

$$n(j) = n_1 + \sum_{i \in \mathcal{P}(j) \setminus \{1\}} \sum_{m=1}^{M} k_{(i,m)} \tag{25}$$

is the total number of failed PEs by node j. Thus, the F-demand of L_m by node j is:

$$\bar{D}_{(j,m)} = \max\{0, \bar{n}(j) - f_{(1,m)} - \sum_{i \in \mathcal{P}(j) \setminus \{1\}} \sum_{m'=1,\cdots,m-1,m+1,\cdots,M} k_{(i,m')}\}. \tag{26}$$

For any given L_m, let i_h, where $h = 1, 2, \cdots, H$ and $H \leq T+1$, be the nodes on $\mathcal{P}(\bar{j})$ such that $\bar{D}_{(i_h,m)} > \bar{D}_{(i,m)}$ for all $i \in \mathcal{P}(a(i_h))$ and $1 < l(i_1) < l(i_2) < \cdots < l(i_H) \leq T+1$. Thus, for any node $j = i_h$ or $j \in \mathcal{P}(i_{h+1}) \setminus (\mathcal{P}(i_h) \cup \{i_{h+1}\})$, the demand of L_m by node j can be determined as: (letting $i_0 = 1$)

$$D_{(j,m)} = \bar{D}_{(i_h,m)}. \tag{27}$$

This revised demand definition considers the truth that it is meaningless for any cumulative demand which is smaller than some demands occurring in earlier stages. The existence of such demand structure is due to the fact that functioning parts on some failed PEs could be used to recover some other failed PEs. Such self-recovery processes might reduce the actual demand on some stages.

Consider the case that $i_H \neq \bar{j}$ for some L_m, i.e., the L_m demand on $\mathcal{P}(\bar{j})$ reaches its maximum before the leaf node $\bar{j}$. With respect to L_m demands, $\mathcal{P}(\bar{j})$ can be replaced with $\mathcal{P}(i_H)$. Part of $\mathcal{P}(\bar{j})$, i.e., from the direct descendant node of i_H to node $\bar{j}$, are vanished with respect to L_m demands. By conducting the same L_m-demand analysis on all other scenarios, there might be more vanished branches and nodes. The scenario tree $\mathcal{T}$ is reduced to the L_m-demand layer, denoted by $\mathcal{T}_m$. Apparently, $\mathcal{T}_m \subseteq \mathcal{T}$. A formal description for how to construct demand layers and how to use this concept in developing and evaluating searching algorithms will be discussed later.

For a given node $j \in \mathcal{V}$ with $l(j) < T+1$, let $\mathbf{x}_j = \{x_{(j,m)} | x_{(j,m)} = x_m^t$, where $x_m^t \in \mathbf{x}^t, m = 1, 2, \cdots, M$ and $l(j) = t\}$ be the notation for LRU-repair decisions on node j. Let $y_j = \{0, 1\}$ be the binary variable such that $y_j = 0$ (if $x_{(j,m)} = 0$ for all $m = 1, 2, \cdots, M$) and $y_j = 1$ (if $x_{(j,m)} > 0$ for some $m \in \{1, 2, \cdots, M\}$). As known, $x_{(j,m)}$ is upper bounded by the number of malfunctioning L_m. In particular, for node 1 the upper bound is given as $x_{(1,m)} \leq g_{(1,m)}$. For all other nodes $j \in \mathcal{V} \setminus \{1\}$ with $l(j) < T+1$, the upper bounds are:

$$x_{(j,m)} \leq g_{(1,m)} + \sum_{i \in \mathcal{P}(j) \setminus \{1\}} \Big(k_{(i,m)} - x_{(a(i),m)} \Big). \tag{28}$$

To satisfy the fleet reliability requirement, the L_m demand by node j should be satisfied as:

$$D_{(j,m)} \leq \sum_{i \in \mathcal{P}(a(j)): j \in \mathcal{V}(h) \text{ where } h \in \mathcal{C}(i, d_{(i,m)})} x_{(i,m)}. \tag{29}$$

Replacing h^t with h_j for $l(j) = t$ gives the formal presentation of MinCostSTM:

$$\text{min:} \quad \sum_{j \in \mathcal{V}: l(j) < T+1} p_j \frac{\left(h_j y_j + \sum_{m=1}^{M} r_m x_{(j,m)} \right)}{(1+\alpha)^{l(j)-1}} \tag{30}$$

$$\text{s.t.:} \quad D_{(j,m)} \leq \sum_{i \in \mathcal{P}(j): j \in \mathcal{V}(h) \text{ where } h \in \mathcal{C}(i, d_{(i,m)})} x_{(i,m)}, \quad \forall j, m \tag{31}$$

$$x_{(j,m)} \leq g_{(1,m)} + \sum_{i \in \mathcal{P}(j) \setminus \{1\}} \left(k_{(i,m)} - x_{(a(i),m)} \right), \quad \forall j, m \tag{32}$$

$$x_{(j,m)} \leq y_j Z, \quad \forall j, m \tag{33}$$

$$x_{(j,m)} \text{ are non-negative integers}, \quad \forall j, m \tag{34}$$

$$y_j \text{ are binary variables}, \quad \forall j, \tag{35}$$

where the objective (30) is to minimize the total expected operating cost. The constraints expressed in (31) describe the fleet reliability requirements and constraints (32) show that the number of repair operations are upper bounded by the number of failed LRUs. Constraints (33) indicate that positive $x_{(j,m)}$ values are available only when positive y_j values are assigned and constraints (34) and (35) demonstrate non-negative $x_{(j,m)}$ and binary y_j, respectively. Note that Z is a constant with a large positive value.

3.2 Polynomial algorithm

In production planning, research on minimizing the total production and inventory cost has a long history. Generally, in order to satisfy demands, which usually dynamically arrive during a finite number of time periods, items are manufactured/ordered with a certain lead time, under a fixed operating cost and a variable cost on each item. In addition, over-manufactured/ordered items and un-satisfied demands might be penalized for overstocking and backorder costs, respectively.

For deterministic production planning problems in which demands in later time periods are also known before or at time one, Wagner and Whitin [33] proved that if the problem had zero initial inventories, unlimited ordering capacities and zero lead times, then there existed an optimal production schedule such that the accumulated order quantity by period t was exactly the accumulated demand by some period τ, where

$0 < t \leq \tau < \infty$. This was called the Wagner-Whitin property. However, with stochastic features, i.e., later demands wouldn't be available until the time comes, Ahmed [34] showed that the Wagner-Whitin property didn't hold if lead times were non-zero. Halman et al. [35] showed that this stochastic version was NP-hard with respect to the number of time periods, even though there were no ordering costs and only two demand events for each time period. For the same stochastic problem, Guan and Miller [36] proved the Production-Path property, which is actually a modified version of Wagner-Whitin property to compromise the stochastic feature of the problem. That is, there existed an optimal production schedule such that the accumulated order quantity by period t was exactly one of the accumulated demand combinations by some period τ, where $0 < t \leq \tau < \infty$. Using this property, they were able to solve the scenario tree version of the problem in polynomial time with respect to the size of the scenario tree. Huang and Kucukyavuz [37] further extended the Production-Path property to more general problems, in which all lead times, ordering, purchasing and inventory costs were random. However, the production planning problems addressed in these papers considered only one type of item and the production/ordering capacities were unlimited over the whole decision horizon.

In MinCostSTM, each PE includes a number of LRUs and a failed PE cannot get back to work until all of the installed LRUs are functioning well, i.e., a PE demand cannot be satisfied until all the associated LRU demands are satisfied. Moreover, in MinCostSTM each newly-failed PE brings one malfunctioning LRU (which can be seen as an LRU demand) and $M-1$ functioning LRUs (which can be cannibalized as functioning LRUs) into the system. MinCostSTM has correlated LRU demands and supplies. On the other hand, in each time period LRU-repair operations are upper-bounded by the number of malfunctioning LRUs at the beginning of the period. Since malfunctioning LRUs are introduced dynamically by the newly-failed PE failures, LRU-repair operations are constrained dynamically. MinCostSTM is a multi-item production planning problem with dynamically-varied production capacities.

In the extended version of Production-Path property (i.e., Proposition 5 in [37]), for any given node j if it has a positive ordering quantity, then the total ordering quantity on $\mathcal{P}(j)$ is equal to the total demand on $\mathcal{P}(i)$, where node i is a node located in one of the subtrees rooted on the nodes in the intersection set of $\mathcal{V}(j)$ and $\mathcal{L}(l(j)+d_j)$. Note that d_j is the lead time for the production decisions made on node j. Since this proposition considers single-item products and unlimited production capacities, in order to coordinate MinCostSTM, where each PE has M LRUs and LRU repair operations are dynamically upper bounded by the number of malfunctioning LRUs, necessary modifications on the property are required.

Recall that if a malfunctioning L_m is sent to the depot for repair on node j, then it will be repaired and returned to the base on all nodes in $\mathcal{C}(j, d_{(j,m)})$, i.e., the nodes in $\mathcal{V}(j)$ with level $l(j)+d_{(j,m)}$. This implies that the L_m-repair operations on node j

can be used to satisfy the L_m demands on any node $h \in \mathcal{V}(i)$, where $i \in \mathcal{C}(j, d_{(j,m)})$. Due to the non-crossover and feasible-solution assumption, the L_m-repair operations on nodes in $\mathcal{P}(a(j))$ should be large enough to satisfy all of L_m demands on nodes in $\cup_{i\in\mathcal{C}(j,d_{(j,m)})}(\mathcal{P}(a(i)) \setminus \mathcal{P}(a(j)))$. Next, Proposition 1 consolidates and modifies the propositions in [33, 36, 37] to reflect the discussions above.

Proposition 1. *For MinCostSTM, there exists an optimal solution* $\pi = \{(y_j, \boldsymbol{x}_j) | \forall j \in \mathcal{V}\}$ *such that if* $x_{(j,m)} > 0$ *for some* $j \in \mathcal{V}$ *and* $m \in \{1, 2, \cdots, M\}$*, then (I)* $y_j = 1$*; (II)* $x_{(j,m)}$ *satisfies the repair capacity constraint* (32)*; and (III) the fleet reliability constraint* (31) *is addressed by:*

$$\sum_{j' \in \mathcal{A}(i,m) \cap \mathcal{P}(j)} x_{(j',m)} = D_{(h,m)} - D_{(a(i),m)}, \tag{36}$$

where $i \in \mathcal{C}(j, d_{(j,m)})$ *and* $h \in \mathcal{V}(i)$*. Note that* $\boldsymbol{x}_j = (x_{(j,1)}, x_{(j,2)}, \cdots, x_{(j,M)})$*.*

Proof. Similarly to the proof of Proposition 5 in [37], this proposition is proved by contradiction. Let node j be some node in $\mathcal{V}$ such that Equation (36) is violated. This means that for all nodes $i \in \mathcal{C}(j, d_{(j,m)})$, where $m \in \{m' | x_{(j,m')} > 0 \text{ and } m' = 1, 2, \cdots, M\}$, there does not exist any node h such that $h \in \mathcal{V}(i)$ and Equation (36) is satisfied. In the following proof, by re-allocating all or a portion of $x_{(j,m)}$, the updated $x_{(j,m)}$ (denoted by $x'_{(j,m)}$) will be zero or satisfy Equation (36). After the re-allocation, it is observed that the fleet reliability constraints are satisfied with in most cases smaller operating costs. (Note that the operating cost won't be increased.) This contradicts the original statement that $(y_j, \mathbf{x}_j) \in \pi$ is an optimal solution to MinCostSTM.

A positive $x_{(j,m)}$ implies $l(j) + d_{(j,m)} \leq T + 1$. If this is not the case, then letting $x_{(j,m)} = 0$ won't cause any problem other than reducing operating costs. For the case of $l(j) + d_{(j,m)} \leq T + 1$, consider a leaf node $i \in \mathcal{C}(j, T + 1 - l(j))$. If $x_{(h,m)} = 0$ for all nodes $h \in \mathcal{P}(a(i)) \setminus \mathcal{P}(j)$, then i is called a *mLeaf* node of node j. The L_m demand on i is satisfied by L_m-repair decisions on $\mathcal{P}(j)$. If $x_{(h,m)} > 0$ for some node $h \in \mathcal{P}(a(i)) \setminus \mathcal{P}(j)$, then let the node with the smallest $l(h)$ such that $l(h) + d_{(h,m)} > l(j) + d_{(j,m)}$ and $l(h) + d_{(h,m)} = l(j) + d_{(j,m)}$ be the *mGRoot* and *mERoot* nodes of node j, respectively. Note that a *mGRoot* (or *mERoot*) node might be shared by more than one node. Let $\hat{\mathcal{B}}(j,m)$ and $\bar{\mathcal{B}}(j,m)$ be the sets including all *mGRoot* and *mERoot* nodes in $\mathcal{V}_j$, respectively. Let $\mathcal{A}'(j,m)$ be the set including all *mLeaf* nodes in $\mathcal{V}_j$. Note that there is at least one non-empty set among $\hat{\mathcal{B}}(j,m)$, $\bar{\mathcal{B}}(j,m)$ and $\mathcal{A}'(j,m)$.

To illustrate these *mERoot*, *mGRoot* and *mLeaf* nodes, Figure 6 depicts an five-level example subtree $\mathcal{V}(\bar{j})$. Let $d_{(\bar{j},m)} = 3$ be the L_m-repair lead time on node $\bar{j}$ and 2 be the L_m-repair lead times on all other nodes in $\mathcal{V}(\bar{j}) \setminus \{\bar{j}\}$. Let nodes $\bar{j}$, h_1 and h_2 be the only nodes, where positive L_m-repair decisions are made; that is the L_m-repair

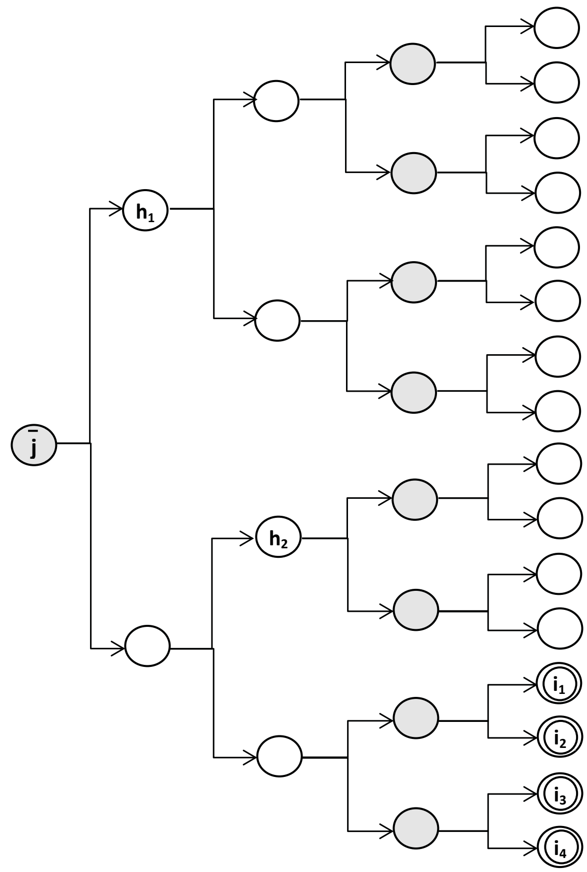


Figure 6: *The mERoot, mGRoot and mLeaf nodes of node* $\bar{j}$ *in the sample problem.*

decisions made on nodes $\bar{j}$, h_1 and h_2 will arrive at the shaded, bold-line-circled and dash-line-circled nodes, respectively. It is easy to observe that nodes i_1, i_2, i_3 and i_4 are the *mLeaf* nodes, and nodes h_1 and h_2 are the *mERoot* and *mGRoot* nodes, respectively. Thus, in Figure 6, $\bar{\mathcal{B}}(\bar{j},m) = \{h_1\}$, $\hat{\mathcal{B}}(\bar{j},m) = \{h_2\}$ and $\mathcal{A}'(\bar{j},m) = \{i_1,i_2,i_3,i_4\}$.

Note that $\hat{\mathcal{B}}(j,m) = \emptyset$ and $\mathcal{A}'(j,m) = \emptyset$ gives $\mathcal{C}(j,d_{(j,m)}) = \cup_{h\in\bar{\mathcal{B}}(j,m)}\mathcal{C}(h,d_{(h,m)})$. Since $p_j = \sum\limits_{h\in\bar{\mathcal{B}}(j,m)} p_h$, allocating $x_{(j,m)}$ to all *mERoot* nodes such as $x'_{(j,m)} = 0$ and $x'_{(h,m)} = x_{(h,m)} + x_{(j,m)}$ for all $h \in \bar{\mathcal{B}}(j,m)$ might reduce the total operating cost by h_j (if $x_{(j,m)}$ is the only positive LRU-repair decision on node j). This zero-valued $x'_{(j,m)}$ satisfies the proposition statement but contradicts with the optimality claim for $x_{(j,m)} > 0$.

If one or none of $\hat{\mathcal{B}}(j,m)$ and $\mathcal{A}'(j,m)$ is empty, let $\mathcal{A}(j,m) = \mathcal{A}'(j,m) \cup \{a(i)|i \in \mathcal{C}(h,d_{(h,m)}), \forall h \in \hat{\mathcal{B}}(j,m)\}$. Let $D_{(\bar{h},m)}$ be the largest L_m demand over all nodes in $\mathcal{A}(j,m)$ and $D_{(\bar{h},m)}$ occurs on node $\bar{h}$. Note that $D_{(\bar{h},m)}$ is also the largest L_m demand over all nodes, which have to be satisfied using the L_m-repair decisions on $\mathcal{P}(j)$. Using the feasible-solution assumption, $x_{(j,m)} > D_{(\bar{h},m)}$. In order to satisfy Equation (36), let $\Delta_{(j,m)} = x_{(j,m)} - D_{(\bar{h},m)}$ be re-allocated amount such that $x'_{(j,m)} = x_{(j,m)} - \Delta_{(\bar{h},m)}$ and $x'_{(h,m)} = x_{(h,m)} + \Delta_{(\bar{h},m)}$ for all $h \in \bar{\mathcal{B}}(j,m)$. Since $p_j > \sum_{h \in \bar{\mathcal{B}}(j,m)} p_h$, the total operating cost is reduced at least (i.e., considering $\alpha = 0$) by a value of $(p_j - \sum_{h \in \bar{\mathcal{B}}(j,m)} p_h)\Delta_{(j,m)} r_m$. This updated $x'_{(j,m)} > 0$ covers the L_m demand on node $\bar{h}$ but contradicts with the optimality claim for $x_{(j,m)} > 0$.

Note that the feasibility of re-allocating $x_{(j,m)}$ and $\Delta_{(j,m)}$ to the *mERoot* nodes in $\bar{\mathcal{B}}(j,m)$ is due to the fact that the un-used L_m-repair capacities (i.e., the un-repaired malfunctioning L_m) can be transferred to all nodes $h \in \bar{\mathcal{B}}(j,m)$ via $\mathcal{P}(a(h)) \setminus \mathcal{P}(j)$, if there is no additional L_m-repair operation on $\mathcal{P}(a(h)) \setminus \mathcal{P}(j)$. □

Proposition 1 indicates that a dynamic programming algorithm, which tries possible LRU repair operations on all non-leaf nodes in $\mathcal{V}$, could be developed and used to determine optimal repair decisions for MinCostSTM. Before this, necessary notations are introduced. For node $j \in \mathcal{V}$, let $v_j(\mathbf{I})$, where $\mathbf{I} = (i_1, i_2, \cdots, i_M)$, be the optimal solution value of the MinCostSTM problem defined on $\mathcal{V}(j)$, where $D_{(i_m,m)}$ are satisfied by the L_m-repair decisions made on $\mathcal{P}(a(j))$ for all nodes $i_m \in \mathbf{I}$. Note that it is possible to have $i_{m_1} \neq i_{m_2}$ and $i_{m_1} = i_{m_2}$, where $i_{m_2}, i_{m_2} \in \mathbf{I}$ and $m_1 \neq m_2$. As two extreme cases, $i_1 = i_2 = \cdots = i_M$ means that $\mathbf{I}$ denotes a single node in $\mathcal{V}$ and $i_{m_1} \neq i_{m_2}$ for all $1 \leq m_1 \neq m_2 \leq M$ means that $\mathbf{I}$ denotes M distinct nodes in $\mathcal{V}$.

Let $u_j(\mathbf{I}, \mathbf{J})$, where $\mathbf{I} = (i_1, i_2, \cdots, i_M)$ and $\mathbf{J} = (j_1, j_2, \cdots, j_M)$, be the objective value for the MinCostSTM problem defined on $\mathcal{V}(j)$ with either $x_{(j,m)} = D_{(j_m,m)} - D_{(i_m,m)}$ or $x_{(j,m)} = 0$ for all $m = 1, 2, \cdots, M$ such that $D_{(i_m,m)}$ are satisfied by the L_m-repair decisions made on $\mathcal{P}(a(j))$ for all $i_m \in \mathbf{I}$. Considering making non-zero L_m-repair decisions on node j, the upper bound for such decisions can be calculated as:

$$g_{(j,m)} = g_{(1,m)} + \sum_{j' \in \mathcal{P}(j')} \left(k_{(j',m)} - x_{(a(j'),m)} \right), \tag{37}$$

i.e., $x_{(j,m)} = D_{(j_m,m)} - D_{(i_m,m)}$ if $0 < D_{(j_m,m)} - D_{(i_m,m)} \leq g_{(j,m)}$. (Note that $x_{(j,m)} < 0$ is not feasible.) On the other hand, $x_{(j,m)} = 0$ is assigned when $D_{(i_m,m)} = D_{(j_m,m)}$ or $D_{(i_m,m)} - D_{(j_m,m)} > g_{(j,m)}$. Thus, the value of $u_j(\mathbf{I}, \mathbf{J})$ can be determined as:

$$u_j(\mathbf{I}, \mathbf{J}) = \begin{cases} p_j \dfrac{\left(h_j + \sum_{m' \in \{1,2,\cdots,M\}} x_{(j,m')} r_{m'} \right)}{(1+\alpha)^{l(j)-1}} + \sum_{j' \in \mathcal{C}(j,1)} v_{j'}(\mathbf{I}), \text{ if for some } m' \\ \qquad 0 < D_{(j_{m'},m')} - D_{(i_{m'},m')} \leq g_{(j,m')} \\ \sum_{j' \in \mathcal{C}(j,1)} v_{j'}(\mathbf{I}), \text{ if for all } m = 1, 2, \cdots, M, \text{ if } D_{(i_m,m)} = D_{(j_m,m)} \\ \qquad \text{or } D_{(j_m,m)} - D_{(i_m,m)} > g_{(j,m)} \end{cases}. \tag{38}$$

For a given L_m and a node j, let $\mathcal{W}(j,m)=\{i|l(j)+d_{(j,m)}\le l(i)<l(j')+d_{(j',m)},\forall j'\in \mathcal{C}(j,1) \text{ and } i\in\mathcal{V}(j')\}$ be the set including all nodes whose L_m demands have to be satisfied by the L_m-repair decisions on $\mathcal{P}(j)$. Using the feasible-solution assumption, for any set of L_m-repair decisions on $\mathcal{P}(a(j))$ the remaining L_m-repair resources on node j would be enough to make the L_m-repair decisions on node j to satisfy the L_m demands on all nodes $i\in\mathcal{W}(j,m)$ for all $m=1,2,\cdots,M$. Thus, the value of $v_j(\mathbf{I})$ can be calculated as:

$$v_j(\mathbf{I})=\min_{j_m\in\{j'|j'\in\mathcal{V}(j) \text{ and } l(j')\ge l(j)+d_{(j,m)}\}:D_{(j_m,m)}-D_{(i_m,m)}\ge 0}\{u_j(\mathbf{I},\mathbf{J})\}. \tag{39}$$

Let $\mathbf{0}$ be the dummy notation for the initial status, i.e., $\mathbf{0}$ represents $\{n_1,\mathbf{f}_1,\mathbf{g}_1,\mathbf{q}_1,\mathbf{s}\}$. Let $v_1(\mathbf{0})$ be the optimal solution value of MinCostSTM. The backwards dynamic programming algorithm, denoted by Recursion, starts from $v_1(\mathbf{0})$. For boundary conditions, if $l(j)+d_{(j,m)}>T+1$ (where node $j\in\mathcal{V}$), then the optimal L_m-repair decision is $x_{(j,m)}=0$. In turn, if $l(j_m)+d_{(j_m,m)}>T+1$ occurs for all $j_m\in\mathbf{J}$, then $u_j(\mathbf{I},\mathbf{J})=0$ is determined for all feasible $\mathbf{I}$. Next, Recursion is summarized.

Algorithm Recursion

[Boundary Conditions]: For any pair of $\mathbf{I}$ and $\mathbf{J}$, if $l(j_m)+d_{(j_m,m)}>T+1$ for all $j_m\in\mathbf{J}$ and $m=1,2,\cdots,M$, then set $u_j(\mathbf{I},\mathbf{J})=0$.

[Recursion Procedure]: For each $u_j(\mathbf{I},\mathbf{J})$, do the calculations in Equation (38):

$$u_j(\mathbf{I},\mathbf{J})=\begin{cases} p_j\dfrac{\left(h_j+\sum_{m'\in\{1,2,\cdots,M\}}x_{(j,m')}r_{m'}\right)}{(1+\alpha)^{l(j)-1}}+\sum_{j'\in\mathcal{C}(j,1)}v_{j'}(\mathbf{I}), & \text{if there is some } m'\in\{1,2,\cdots,M\} \text{ with } 0<D_{(j_{m'},m')}-D_{(i_{m'},m')}\le g_{(j,m')} \\ \sum_{j'\in\mathcal{C}(j,1)}v_{j'}(\mathbf{I}), & \text{if for all } m=1,2,\cdots,M, \text{ it is occurred such that either } D_{(i_m,m)}=D_{(j_m,m)} \text{ or } D_{(j_m,m)}-D_{(i_m,m)}>g_{(j,m)} \end{cases}.$$

For any $j'\in\mathcal{C}(j,1)$, do the following calculations as described in Equation (39):

$$v_{j'}(\mathbf{I})=\min_{i'_m\in\{j''|j''\in\mathcal{V}(j') \text{ and } l(j'')\ge l(j')+d_{(j',m)}\}:D_{(i'_m,m)}-D_{(i_m,m)}\ge 0}\{u_{j'}(\mathbf{I},\mathbf{I}')\},$$

where $\mathbf{I}'=(i'_1,i'_2,\cdots,i'_M)$.

[Optimal Solution]: Recall that the LRU demands on node 1 is $\mathbf{D}_0=(0,0,\cdots,0)$.

$$v_1(\mathbf{0})=\min_{j_m\in\{j'|j'\in\mathcal{V} \text{ and } l(j')\ge 1+d_{(1,m)}\}:D_{(j_m,m)}\ge 0}\{u_1(\mathbf{0},\mathbf{J})\}. \tag{40}$$

Theorem 1. *Recursion finds an optimal solution to MinCostSTM in* $O(|\mathcal{V}|^{2M+2})$ *time, where* $|\mathcal{V}| = \sum_{t=0}^{T+1} \Gamma^t$ *is the total number of nodes on the scenario tree presentation of MinCostSTM. Note that for any given* K *and* M *in MinCostSTM,* Γ *can be determined using the procedures described in Equations (21) and (22).*

Proof. The correctness follows Proposition 1 and the discussions above. Regarding the complexity, the total number of $u_j(\mathbf{I},\mathbf{J})$ is upper bounded by the production of the number of j, $\mathbf{I}$ and $\mathbf{J}$, whose maximum values are $|\mathcal{V}|$, $|\mathcal{V}|^M$ and $|\mathcal{V}|^M$, respectively. Since the calculation of each $u_j(\mathbf{I},\mathbf{J})$ requires to evaluate exactly $|\mathcal{C}(j,1)|$ number of $v_{j'}(\mathbf{I})$, i.e., one evaluation for each node $j' \in \mathcal{C}(j,1)$, the total calculations require $O(|\mathcal{V}| \times |\mathcal{V}|^M \times |\mathcal{V}|^M \times |\mathcal{C}(j,1)|) < O(|\mathcal{V}|^{2M+2})$ computing times, where $|\mathcal{C}(j,1)| < |\mathcal{V}|$ is the number of nodes in $\mathcal{C}(j,1)$. Since $u_j(\mathbf{I},\mathbf{J})$ computing dominates Recursion, the overall run time is $O(|\mathcal{V}|^{2M+2})$. $\square$

3.3 Run time analysis

In MinCostSTM, a PE-failure (say $\mathbf{k}_j = (k_{(j,1)}, k_{(j,2)}, \cdots, k_{(j,M)})$) introduces into the maintenance system $k_{(j,m)}$ and $\sum_{m'=1,\cdots,m-1,m+1,\cdots,M}^{M} k_{(j,m')}$ number of malfunctioning and functioning L_m, respectively. There is no doubt that these functioning L_m can be used to recover failed PEs via cannibalization. As such, the L_m demands can be reduced without using stocked L_m. This L_m-demand reduction might cause irregular demands on L_m, e.g., $\bar{D}_{(j,m)} < \bar{D}_{(a(j),m)}$ (for L_m F-demands) and $D_{(j,m)} = D_{(a(j),m)}$ (for L_m demands) for some node $j \in \mathcal{V}$.

Consider a leaf node $j \in \mathcal{V}$, where $D_{(j,m)} = D_{(a(j),m)}$ for some m. Since $D_{(j,m)}$ is satisfied once $D_{(a(j),m)}$ is satisfied, eliminating node j from $\mathcal{V}$ makes no difference with respect to L_m-repair decisions or L_m-demand sanctifications. If leaf node j is the only descendant node of node $a(j)$, then after node j is eliminated node $a(j)$ become another leaf node. Another round of elimination might be applied on $a(j)$. By repeatedly doing all such eliminations from the node with the highest level on each scenario/path, the L_m-demand layer $\mathcal{T}_m$ is obtained. Let $\mathcal{V}_m$ denote the set of nodes on $\mathcal{T}_m$. By merging all $\mathcal{T}_m$, a residual tree (denoted by $\mathcal{T}_R = \cup_{m=1}^{M} \mathcal{T}_m$) is obtained. Clearly, $\mathcal{T}_R \subseteq \mathcal{T}$. Similarly, letting $\mathcal{V}_R$ include all nodes on $\mathcal{T}_R$ gives $\mathcal{V}_R = \cup_{m=1}^{M} \mathcal{V}_m$. Clearly, $\mathcal{V}_R \subseteq \mathcal{V}$ and $|\mathcal{V}_R| \leq |\mathcal{V}|$. Similarly, let $\mathcal{T}_m(j)$ be the L_m-demand layer of subtree $\mathcal{T}(j)$ and let $\mathcal{T}_R(j)$ be the residual tree of subtree $\mathcal{T}(j)$.

Using these demand layers and residual tree, the calculation of optimal solution value of the $\mathcal{T}_R(j)$-based MinCostSTM problem, i.e., Equation (39), should be modified as:

$$v_j(\mathbf{I}) = \min_{j_m \in \{j' | j' \in \mathcal{V}_m(j) \text{ and } l(j') \geq l(j) + d_{(j,m)}\}: D_{(j_m,m)} - D_{(i_m,m)} \geq 0} \{u_j(\mathbf{I},\mathbf{J})\}. \quad (41)$$

It might happen for some node j and L_m, $\mathcal{V}_m(j) = \emptyset$. In this case, $j_m = i_m$ is assigned and therefore $x_{(j,m)} = 0$. Accordingly, the $v_{j'}(\mathbf{I})$ and $v_1(\mathbf{0})$ calculations in Recursion are modified to search for nodes i'_m and j_m on $\mathcal{V}_m(j')$ and $\mathcal{V}_m$, respectively. The latter one requires the following modifications on Equation (40):

$$v_1(\mathbf{0}) = \min_{j_m \in \{j' | j' \in \mathcal{V}_m \text{ and } l(j') \geq 1 + d_{(1,m)}\} : D_{(j_m,m)} \geq 0} \{u_1(\mathbf{0}, \mathbf{J})\}. \tag{42}$$

Similarly, the calculations of $u_j(\mathbf{I},\mathbf{J})$ in Equation (38) should be modified as:

$$u_j(\mathbf{I},\mathbf{J}) = \begin{cases} p_j \dfrac{\left(h_j + \sum\limits_{m' \in \{1,2,\cdots,M\}} x_{(j,m')} r_{m'}\right)}{(1+\alpha)^{l(j)-1}} + \sum\limits_{j' \in \mathcal{C}_R(j,1)} v_{j'}(\mathbf{I}), \text{ if for some } m' \\ \qquad 0 < D_{(j_{m'},m')} - D_{(i_{m'},m')} \leq g_{(j,m')} \\ \sum\limits_{j' \in \mathcal{C}_R(j,1)} v_{j'}(\mathbf{I}), \text{ if for all } m = 1,2,\cdots,M, \text{ if } D_{(i_m,m)} = D_{(j_m,m)} \\ \qquad \text{or } D_{(j_m,m)} - D_{(i_m,m)} > g_{(j,m)} \end{cases}, \tag{43}$$

where $\mathcal{C}_R(j,1)$ is the set of descendant nodes of j on $\mathcal{T}_R$ and $\mathcal{C}_R(j,1) \subseteq \mathcal{C}(j,1)$.

To illustrate how the idea of demand layers and residual tree reduces run times, a four-period sample problem is used. In the problem, all PEs are made up of two different LRUs, LRU-1 and LRU-2. Initially, there are no spare or due-in LRUs for either LRU-1 or LRU-2 (i.e., $s_1 = s_2 = q_1 = q_2 = 0$), and there are two failed PEs with one malfunctioning LRU-1 and two malfunctioning LRU-2 as in Table 1.

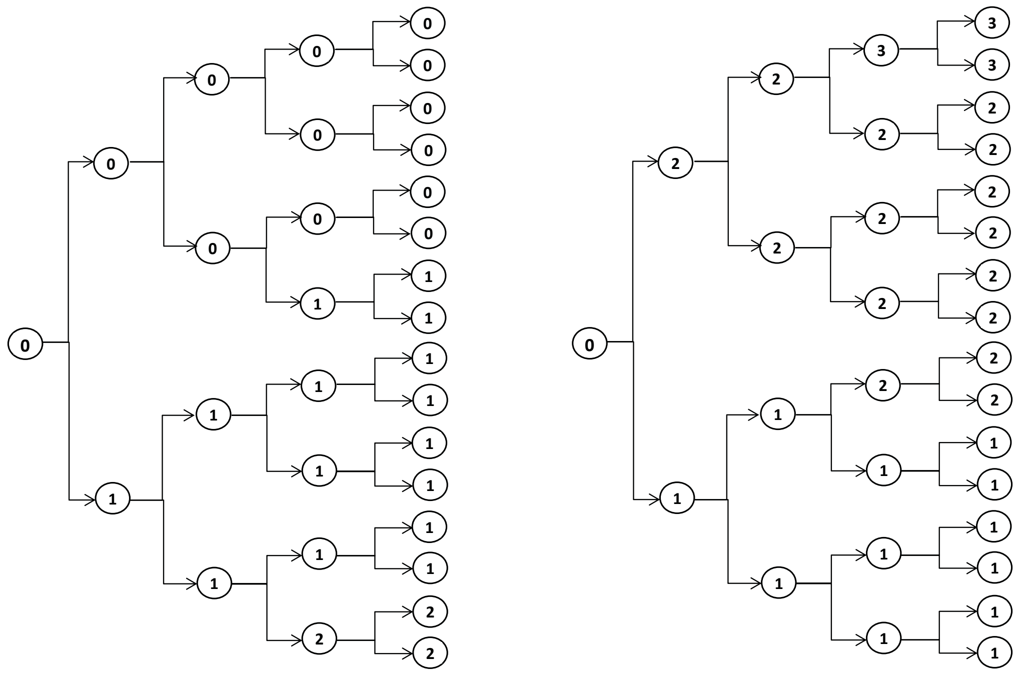
Table 1: *The basic information of the sample problem.*

Items	Number of Failed/Malfunctioning Items		
	Initial Status	Type-1 PE-failure	Type-2 PE-failure
PE	2	1	1
LRU-1	1	1	0
LRU-2	2	0	1

As showed in the table, there are two types of PE-failures: i.e., Type-1 PE failure has one failed PE with a malfunctioning LRU-1 on it, while Type-2 PE failure has one failed PE with a malfunctioning LRU-2 on it. Note that since this problem is used to compare scenario tree, residual tree and demand layers, unrelated information (e.g., the operating costs, lead times, and PE-failure probabilities) are not presented.

Let $A = 0.5$ be the designated level for fleet reliabilities. Since in each time period PE failures introduce exactly one failed PE, the number of PEs (which are required to be recovered at the end of each time period) are two (by time 2), two (by time 3), three (by time 4) and three (by time 5). As such, the demands for LRU-1 and LRU-2 can be calculated easily, which are presented on the scenario trees depicted by Figure 7 (a) and (b), respectively.

Note that both trees share the same tree structure, which is the tree structure of the scenario tree presentation of the sample problem. On each non-leaf node, the upper and lower branches represent Type-1 and Type-2 PE failures, respectively. It is observed that on almost all nodes the demands for LRU-1 are less than the demands for LRU-2. Apparently, this is due to the initial status that the two failed PEs have one functioning LRU-1 and zero functioning LRU-2.

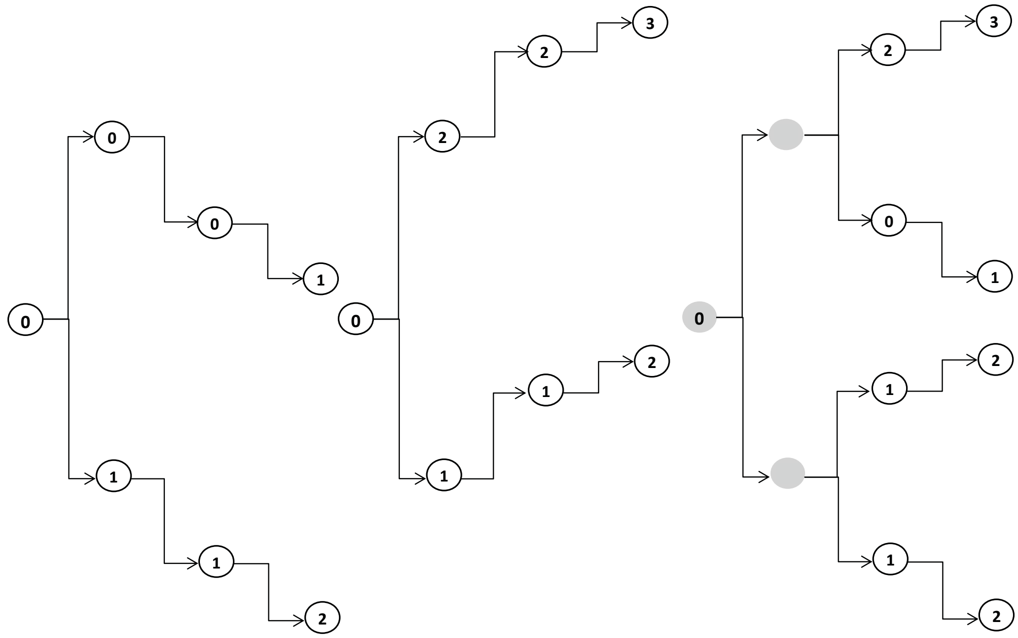


(a) Demands of LRU-1 **(b) Demands of LRU-2**

Figure 7: *The LRU demands of the sample problem.*

Following the above node elimination procedures, the LRU-1 and LRU-2 demand trees in Figure 7 (a) and (b) are reduced to the LRU-1 and LRU-2 demand layers in Figure 8 (a) and (b), respectively. It is observed that Figure 8 (c) is the residue tree, which is the combination of the two demand layers: Figure 8 (a) and (b).

In Figure 8 (c), the nodes, which are depicted on both demand layers (Figure 8 (a) and (b)), are shaded. The zero value marked on the root node indicates zero demands for both LRUs. For the other two shaded nodes, since the demands for LRU-1 and LRU-2 are different, no demand value is marked. The solid-line and dash-line circled



(a) Layer of LRU-1 demands (b) Layer of LRU-2 demands (b) Residual tree

(a) Layer of LRU-1 demands **(b) Layer of LRU-2 demands** **(c) Residual tree**

Figure 8: *The demand layers and residual tree of the sample problem.*

nodes are the nodes depicted on LRU-1 and LRU-2 demand layers (Figure 8 (a) and (b)), respectively. The marked values in these non-shaded nodes are the associated demands for LRU-1 and LRU-2.

It is observed that the number of nodes are reduced from 31 (the original scenario tree in Figure 7 (a) and (b)) to 7 (the LRU demand layers in Figure 8 (a) and (b)) and 11 (the residual tree in Figure 8 (c)). Using the original scenario tree (Figure 7 (a) and (b)), the estimated run time for Recursion would be $31^{2+2\times2} = 887,503,681$. However, if Recursion uses the residual tree (Figure 8 (c)) and the two LRU demand layers (Figure 8 (a) and (b)), i.e., replacing Equations (38), (39) and (40) with Equations (43), (41) and (42), respectively, the run time can be estimated by $11^2 \times 7^{2\times2} = 2,522$. Clearly, Recursion's performance is substantially improved.

4 Approaches to MaxRho

In MaxRho, the goal is determine a set of repair operations to maximize the fleet reliability level such that the total operating cost is within the limited budget. However, in most military situations, if repair operations have direct effects on fleet reliabilities, then there won't be *strict* budget limit on operating costs. For example, in CAF, *contingency funds* are always available for unexpected maintenance/repair operations as long as there is a need. Under this consideration, the scenario tree presentation of MaxRho (denoted by MaxRhoSTM) is a means of determining over a finite number of decision periods the best repair operations such that the fleet reliability level is maximized and the *expected* value of the total operating cost is under the limited budget. Note that MinCostSTM presented in Equations (30), (31), (32), (33), (34) and (35) is to minimize the *expected* value of the total operating cost with the fleet reliability satisfying the designated level. It is obvious that for the same maintenance system, MaxRhoSTM and MinCostSTM share the same scenario tree structure.

Let $D_{(j,m)}(\rho)$ denote the L_m demand by node $j \in \mathcal{V}$ with respect to required fleet reliability level ρ, where $m = 1, 2, \cdots, M$ and $0 \leq \rho \leq 1$. Note that the value of $D_{(j,m)}$ used in MinCostSTM is obtained based on the required reliability level A, i.e., $D_{(j,m)} = D_{(j,m)}(A)$. Using the calculation procedures described in Section 3.1, $D_{(j,m)}(\rho)$ can be easily obtained for any given ρ value. Note that these calculations cannot be presented explicitly as general mathematical formulas, and therefore any mathematical programming problem with $D_{(j,m)}(\rho)$ as intermediate variables is difficult to solve. In the following formulation, MaxRhoSTM is to maximize ρ and includes $D_{(j,m)}(\rho)$ as parts of the constraints. This shows that MaxRhoSTM is mathematically intractable. Special solution methodologies will be developed for it.

$$\text{max:} \quad \rho \tag{44}$$

$$\text{s.t.:} \quad D_{(j,m)}(\rho) \leq \sum_{i \in \mathcal{P}(a(j)): j \in \mathcal{V}(k) \text{ where } k \in \mathcal{C}(i, d_{(i,m)})} x_{(i,m)}, \quad \forall j, m \tag{45}$$

$$\sum_{j \in \mathcal{V}} p_j \frac{\left(h_j y_j + \sum_{m=1}^{M} r_m x_{(j,m)} \right)}{(1+\alpha)^{l(j)-1}} \leq B \tag{46}$$

$$x_{(j,m)} \leq g_{(1,m)} + \sum_{i \in \mathcal{P}(a(j))} \left(k_{(i,m)} - x_{(i,m)} \right), \quad \forall j, m \tag{47}$$

$$x_{(j,m)} \leq y_j Z, \quad \forall j, m \tag{48}$$

$$0 \leq \rho \leq 1 \tag{49}$$

$$x_{(j,m)} \text{ are non-negative integers}, \quad \forall j, m \tag{50}$$

$$y_j \text{ are binary variables}, \quad \forall j, \tag{51}$$

where (44) shows that the goal is to maximize the fleet reliability level and (49) constrains the level in a feasible range $[0,1]$. Constraints (45) and (46) describe the demand constraints (which reflect the designated fleet reliability level) and the budget constraint (which requires the expected value of the total operating cost), respectively. All other constraints (47), (48), (50) and (51) have the same meanings as in MinCostSTM.

As discussed above, it is unlikely that MaxRhoSTM can be solved using general optimization approaches. In this case, combinatorial optimization approaches (e.g., dynamic programming algorithms) might be useful in solving MaxRhoSTM. Generally, dynamic programming algorithms enumerate and compare all candidate solutions to find out the solution with the largest/smallest objective values as the optimal solution to maximization/minimization problems. As a requirement, there should be a finite number of value options for decision variables. In MaxRhoSTM, however, $\rho \in [0,1]$ gives unlimited value options for ρ. It is unlikely that MaxRhoSTM can be optimally solved using any dynamic programming algorithm. In the remaining of this section, the focus is to develop approximation algorithms for MaxRhoSTM.

The approximation algorithm can be structured as a general binary search procedure, where Recursion is repeatedly called to solve a series of MinCostSTM problems. In these MinCostSTM problems, designated reliability levels are dynamically updated. Let MinCostSTM(ρ) denote the MinCostSTM problem, whose fleet reliability level is designated to be ρ. In particular, MinCostSTM(A) is an alternative of MinCostSTM as the required fleet reliability level for MinCostSTM is A. In each iteration of the binary search, an on-hand ρ value is used to determine a particular MinCostSTM(ρ) and Recursion is called to find the solution. If a feasible solution is found and the objective value is within the limited budget (i.e., the operating budget constrained in (46) is large enough to provide the fleet reliability leve ρ), then the result is *acceptable* and ρ is updated to a larger value for next iteration; otherwise it concludes that the result is *non-acceptable* (i.e., the operating budget constrained in (46) is not enough to provide the fleet reliability level ρ) and ρ is updated to a smaller value for next iteration.

Let ρ^* and ρ^ε be the optimal and approximation solution values of MaxRhoSTM, respectively. Since ρ^* is maximized in MaxRhoSTM, $\rho^\varepsilon \leq \rho^*$ is obvious. In general, there are two ways to measure the quality of ρ^ε: (1) the relative measure $\rho^{\varepsilon t} \geq \varepsilon_t^{-1} \times \rho^*$ (where $\varepsilon_t > 1$ is the *ratio* indicator and $\rho^* \in [\rho^{\varepsilon t}, \varepsilon_t \times \rho^{\varepsilon t}]$ is determined) and (2) the absolute measure $\rho^{\varepsilon n} \geq \rho^* - \varepsilon_n$ (where $\varepsilon_n > 0$ is the *range* indicator and $\rho^* \in [\rho^{\varepsilon n}, \rho^{\varepsilon n} + \varepsilon_n]$ is determined). Based on this, two approximation algorithms, BinRecT and BinRecN, are developed to determine the approximation intervals for ρ with respect to ε_t and ε_n, respectively.

Let U and L be the upper and lower bound of ρ^*, respectively. Let $t = \frac{U}{L}$ and $n = U - L$ be the bound ratio and range, respectively. Using the assumption that there is at least one feasible solution to MaxRhoSTM, it is for sure that there is at least one failed PE which could be recovered using the limited budget. This gives the initial lower bound $L = \frac{1}{T\bar{K}+N}$. Using the initial upper bound $U = 1$, the initial bound ratio and range are $t = T\bar{K} + N$ and $n = \frac{T\bar{K}+N-1}{T\bar{K}+N}$, respectively. In BinRecT and BinRecN, t and n are repeatedly updated and compared to ε_t and ε_n, respectively. BinRecT and BinRecN stop as $t \leq \varepsilon_t$ and $n \leq \varepsilon_n$, respectively. First, BinRecT is presented.

Algorithm BinRecT

[Preparation]: Initialize the searching parameters and determine the stop criteria.

- Set $L = \frac{1}{T\bar{K}+N}$, $U = 1$, $t = T\bar{K} + N$ and $\rho(\varepsilon_t) = L$.
- Determine the value of ε_t, where ε_t is required to be in $(1,t)$.

[BinarySearch]: If $t \leq \varepsilon_t$, then go to [FinalCheck]. Otherwise do:

- Set $\rho = \frac{L+U}{2}$ and determine $D_{(j,m)}(\rho)$ for all $j \in \mathcal{V}$ and $m = 1,2,\cdots,M$.
- Run Recursion to solve MinCostSTM(ρ).
  - If the returned result is non-acceptable, then set $U = \rho$.
  - If the returned result is acceptable, then set $L = \rho$ and $\rho(\varepsilon_t) = L$.
- Calculate $t = \frac{U}{L}$ and go to [BinarySearch].

[FinalCheck]: Determine $D_{(j,m)}(U)$ for all $j \in \mathcal{V}$ and $m = 1,2,\cdots,M$, and run Recursion to solve MinCostSTM(U).

- If the returned result is non-acceptable, then go to [solution].
- If the returned result is acceptable, then set $\rho(\varepsilon_t) = U$.

[Solution]: Let $\rho^{\varepsilon_t} = \rho(\varepsilon_t)$ be the found fleet reliability level and trace back to obtain the corresponding repair decisions.

Theorem 2. *For MaxRhoSTM, BinRecT can find an* $(\varepsilon_t^{-1} - \text{ratio})$*-approximation solution such as* $\rho^{\varepsilon_t} \geq \varepsilon_t^{-1} \times \rho^*$ *in* $O(\lceil \log_{\varepsilon_t}(T\bar{K}+N) \rceil |\mathcal{V}|^{2M+2})$ *time.*

Proof. In [Preparation], the initial setting for $\rho(\varepsilon_t) = \frac{1}{T\bar{K}+N}$ is due to the assumption that MaxRhoSTM has at least one feasible solution which provides the reliability level at least $\frac{1}{T\bar{K}+N}$. In [BinarySearch], the bound interval is cut into half after each iteration, by either increasing L or decreasing U. $\rho(\varepsilon_t)$ is updated only when an acceptable result is returned. After [FinalCheck], the bound interval is featured such as $\rho^* \in [\rho(\varepsilon_t), \varepsilon_t \times \rho(\varepsilon_t)]$; that is the found solution $\rho^{\varepsilon_t} = \rho(\varepsilon_t) \geq \varepsilon_t^{-1} \times \rho^*$.

Considering time complexity, in each iteration the run time is dominated by Recursion, which runs in $O(|\mathcal{V}|^{2M+2})$ time to solve MinCostSTM(ρ). Since BinRecT conducts binary search and calls Recursion at most $\lceil \log_{\varepsilon_t}(T\bar{K}+N) \rceil + 1$ times (including [FianlCheck]), the overall run time is $O(\lceil \log_{\varepsilon_t}(T\bar{K}+N) \rceil |\mathcal{V}|^{2M+2})$. □

In BinRecT, the qualities of found solution, i.e., the value of approximated bound ratio (t^*), can be pre-estimated by setting the values of ε_t. It is obvious that $\varepsilon_t = 1$ gives $\rho^{\varepsilon_t} \geq \varepsilon_t^{-1} \times \rho^* = \rho^*$. This implies that $\rho^{\varepsilon_t} = \rho^*$ and therefore BinRecT is an optimal algorithm. However, such an optimal solution, which requires a pre-determined approximation parameter $\varepsilon_t = 1$, requires infinite run time for BinRecT, i.e., $\log_1(T\bar{K} + N) = \infty$. It is impossible. This shows that using BinRecT solution qualities can be bought by run times but MaxRhoSTM cannot be solved optimally. Next, BinRecN is designed under a similar algorithm structure. The above solution accuracy and algorithm run time arguments are valid for BinRecN as well.

Algorithm BinRecN

[Preparation]: Initialize the searching parameters and determine the stop criteria.

- Set $L = \frac{1}{T\bar{K}+N}$, $U = 1$, $n = \frac{T\bar{K}+N-1}{T\bar{K}+N}$ and $\rho(\varepsilon_n) = L$.
- Determine the value of ε_n, where ε_n is required to be in $(0, n)$.

[BinarySearch]: If $n \leq \varepsilon_n$, then go to [FinalCheck]. Otherwise do:

- Set $\rho = \frac{L+U}{2}$ and determine $D_{(j,m)}(\rho)$ for all $j \in \mathcal{V}$ and $m = 1, 2, \cdots, M$.
- Run Recursion to solve MinCostSTM(ρ).
  - If the returned result is non-acceptable, then set $U = \rho$.
  - If the returned result is acceptable, then set $L = \rho$ and $\rho(\varepsilon_n) = L$.
- Calculate $n = U - L$ and go to [BinarySearch].

[FinalCheck]: Determine $D_{(j,m)}(U)$ for all $j \in \mathcal{V}$ and $m = 1, 2, \cdots, M$, and run Recursion to solve MinCostSTM(U).

- If the returned result is non-acceptable, then go to [solution].
- If the returned result is acceptable, then set $\rho(\varepsilon_n) = U$.

[Solution]: Let $\rho^{\varepsilon_n} = \rho(\varepsilon_n)$ be the found availability level and trace back to obtain the corresponding decision strategies.

Theorem 3. *For MaxRhoSTM, BinRecN can find an* ($\varepsilon_n - \text{range}$)*-approximation solution such as* $\rho^{\varepsilon_n} > \rho^* - \varepsilon_n$ *in* $O(\lceil \log_{\varepsilon_n}\left(\frac{T\bar{K}+N-1}{T\bar{K}+N}\right)\rceil |\mathcal{V}|^{2M+2})$ *time.*

Proof. Similarly to the proof of Theorem 2, BinRecN exits with $\rho^* \in [\rho(\varepsilon_n), \rho(\varepsilon_n) + \varepsilon_n)$. This gives the found solution $\rho^{\varepsilon_n} = \rho(\varepsilon_n) > \rho^* - \varepsilon_n$. In order to reduce the range from $\frac{T\bar{K}+N-1}{T\bar{K}+N}$ to ε_n, BinRecN calls Recursion at most $\lceil \log_{\varepsilon_n}\left(\frac{T\bar{K}+N-1}{T\bar{K}+N}\right)\rceil$ times. Since in each iteration Recursion dominates the searching operation and Recursion runs $O(|\mathcal{V}|^{2M+2})$ time for each MinCostSTM(ρ), the overall time complexity of BinRecN is $O(\lceil \log_{\varepsilon_n}\left(\frac{T\bar{K}+N-1}{T\bar{K}+N}\right)\rceil |\mathcal{V}|^{2M+2})$. □

5 Final remarks

In this report, an analytical model, MaxRho was developed for multi-stage repair decision making in a single-base, single-depot military maintenance network, where cannibalization operations are allowed and well-managed in the operating base and repair operations are conducted in the depot. The goal was to maximize the overall fleet reliability level over a finite decision horizon, while the total operating cost is within the limited budget. Taking into account most real situations, MaxRho included two uncertain factors: the independent failures and the out-of-control repair operations, i.e., the times and the costs. Under peaceful conditions operating managers are encouraged to dynamically find out which decision is best, MaxRho can be used to determine the most appropriated repair operations for any determined time periods such that the total operating costs are within the limited budget and the fleet reliabilities achieve the designated level.

To solve MaxRho, a complementary minimization problem, MinCost was developed, where the goal was to minimize the total operating costs with constraints on fleet reliabilities. Instead of solving MinCost directly, this report first re-formulated MinCost as a deterministic problem on a scenario tree (MinCostSTM), and then developed a dynamic programming algorithm (Recursion) for MinCostSTM. Finally, it was proved that Recursion could find an optimal solution to MinCostSTM in a polynomial time with respect to the size of the scenario tree. In addition, by conducting more detailed analysis on the scenario tree, it was showed via a sample problem that Recursion could run much faster than the claimed run time.

Instead of directly dealing with MaxRho, this report developed two approximation algorithms, BinRecT and BinRecN to solve MaxRhoSTM (the scenario tree presentation and the deterministic equivalence of MaxRho). Both algorithms implemented binary search to iteratively narrow the intervals, which included the unknown optimal solution value of MaxRhoSTM. In BinRecT and BinRecN, the found solutions were within the relative (ratio) and absolute (range) measured neighborhoods of the unknown optimum, respectively. Moreover, it was proved that the run times of BinRecT and BinRecN would be longer if the ratio and range neighborhood parameters were designated for more accurate solutions.

This report conducted theoretical studies for making detailed multi-stage repair decisions in military maintenance systems with cannibalization. As a future work, more applicable solution approaches for close-to-real problems deserve researchers' efforts, i.e., the GAMS-CPLEX combined approach to large-scale stochastic optimization problems. Other research directions could be to extend the maintenance system from a single-based network to multi-based, and/or to allow more indenture levels in identifying, repairing and cannibalizing failed parts.

References

[1] United States General Accounting Office (2001), Report to the Chairman, Subcommittee on National Security, Veterans Affairs, and International Relations, Committee on Government Reform, House of Representatives: Military Aircraft Services Need Strategies to Reduce Cannibalizations. GAO-02-86. November.

[2] DoD (2013), DoD Maintenance 2013 Fact Book, (Technical Report) Office of the Assistant Secretary of Defense (Logistics and Materiel Readiness).

[3] Sherbrooke, G. C. (1968), METRIC: A Multi-echelon Technique for Recoverable Item Control, *Operations Research*, 16, 122–141.

[4] Muckstadt, J. A. (1973), A model for a Multi-item, Multi-echelon, Multi-indenture Inventory System, *Management Science*, 20(4), 472–481.

[5] Graves, S. (1985), A multi-echelon inventory model for a repairable item with one-for-one replenishment, *Management Science*, 31(10), 1247–1256.

[6] Diaz, A. and Fu, M. (1997), Models for multi-echelon repairable item inventory systems with limited repair capacity, *European Journal of Operational Research*, 97(3), 480–492.

[7] Sleptchenko, A., van der Heijden, M., and van Harten, A. (2002), Effects of finite repair capacity in multi-echelon, multi-indenture service part supply systems, *International Journal of Production Economics*, 79(3), 209–230.

[8] Zhang, R. (2013), Detailed Cannibalization Decision Making for Maintenance Systems in the Military Context, (DRDC CORA TM 2013–228) DRDC – Centre for Operational Research and Analysis.

[9] Pugliese, D. (2009), Canadian Army ERYX Missile System Being Cannibalized to Keep It Going Until 2016 (online), http://blogs.ottawacitizen.com/2009/12/14/canadian-army-eryx-missile-system-being-cannibalized-to-keep-it-going-until-2016 (Access Date: 4 June 2013 at 1137 EDT). Ottawa Citizen. Defence Watch section.

[10] Simon, R. M. (1970), Optimal Cannibalization Policies for Multicomponent Systems, *SIAM Journal on Applied Mathematics*, 19(4), 700–711.

[11] Defourny, B., Ernst, D., and Wehenkel, L. (2012), Multistage Stochastic Programming: A Scenario Tree Based Approach to Planning under Uncertainty, In Sucar, L. E., Morales, E. F., and Hoey, J., (Eds.), *Decision Theory Models for*

Applications in Artificial Intelligence: Concepts and Solutions, pp. 97–143, IGI Global.

[12] Marti, K. (2005), Stochastic Optimization Methods, Springer, Berlin.

[13] Barlow, R. E. and Proschan, F. (1965), Mathematical Theory of Reliability, John Wiley & Sons, Inc., New York.

[14] McCall, J. J. (1965), Maintenance Policies for Stochastically Failing Equipment: A Survey, *Management Science*, 11(5), 493–524.

[15] Osaki, S. and Nakagawa, T. (1976), Bibliography for Reliability and Availability of Stochastic Systems, *IEEE Transactions on Reliability*, 25(4), 284–287.

[16] Pierskalla, W. P. and Voelker, J. A. (1976), A survey of Maintenance Models: The Control and Surveillance of Deteriorating Systems, *Naval Research Logistics Quarterly*, 23(3), 353–388.

[17] Sherif, Y. S. and Smith, M. L. (1981), Optimal Maintenance Models for Systems Subject to Failures - A Review, *Naval Research Logistics Quarterly*, 28, 47–74.

[18] Valdez-Flores, C. and Feldman, R. M. (1989), A Survey of Preventive Maintenance Models for Stochastically Deteriorating Single-unit Systems, *Naval Research Logistics*, 36(4), 419–446.

[19] Cho, D. I. and Parlar, M. (1991), A Survey of Maintenance Models for Multi-unit Systems, *European Journal of Operational Research*, 51(1), 1–23.

[20] Dekker, R. (1996), Applications of Maintenance Optimization Models: A Review and Analysis, *Reliability Engineering & System Safety*, 51(3), 229–240.

[21] Wang, H. (2002), A Survey of Maintenance Policies of Deteriorating Systems, *European Journal of Operational Research*, 139(3), 469–489.

[22] Nicolai, R. P. and Dekker, R. (2008), Complex System Maintenance Handbook, Springer Series in Reliability Engineering, Ch. Optimal Maintenance of Multi-component Systems: A Review, pp. 263–286, Springer London.

[23] Basten, R. J. I., Schutten, J. M. J., and der Heijden, M. C. V. (2009), An Efficient Model Formulation for Level of Repair Analysis, *Annals of Operations Research*, 172(1), 119–142.

[24] Basten, R. J. I., der Heijden, M. C. V., and Schutten, J. M. J. (2012), Practical Extensions to a Minimum Cost Flow Model for Level of Repair Analysis, *European Journal of Operational Research*, 211(32), 333–342.

[25] Tysseland, B. E. (2009), Spare Parts Optimization Process and Results: OPUS10 Cases in the Norwegian Defence, *International Journal of Physical Distribution & Logistics Management*, 39(1), 8–27.

[26] Tang, L. C. and Liu, X. (2011), Spares Provisioning for Repairable Systems under Fleet Expansion, *IEEE Proceedings on 2011 Annual Reliability and Maintainability Symposium (RAMS)*, pp. 1–5.

[27] Basten, R. J. I., der Heijden, M. C. V., and Schutten, J. M. J. (2012), Joint Optimization of Level of Repair Analysis and Spare Parts Stocks, *European Journal of Operational Research*, 222(3), 474–483.

[28] Puterman, M. L. (1994), Markov Decision Processes, John Wiley & Sons, Hoboken, 1st edition.

[29] Ruszczynski, A. and Shapiro, A. (2003), Handbooks in Operations Research and Management Science (vol. 10): Stochastic Programming, Elsevier Science B.V., 1st edition.

[30] Bondorf, J. P., Donangelo, R., and Mishustin, I. N. (1985), Statistical Multifragmentation of Nuclei: (I). Formulation of the Model, *Nuclear Physics A*, 443(2), 321–347.

[31] Bondorf, J. P., Donangelo, R., Mishustin, I. N., and Schulz, H. (1985), Statistical Multifragmentation of Nuclei: (II). Application of the Model to Finite Nuclei Disassembly, *Nuclear Physics A*, 444(3), 460–476.

[32] Desesquelles, P. (2002), Calculation of the Number of Partitions with Constraints on the Fragment Size, *Physical Review C*, 65(3), 034603.

[33] Wagner, H. M. and Whitin, T. M. (1958), Dynamic Version of the Economic Lot Size Model, *Management Science*, 5(1), 89–96.

[34] Ahmed, S., King, A. J., and Parija, G. (2003), A Multi-stage Stochastic Integer Programming Approach for Capacity Expansion under Uncertainty, *Journal of Global Optimization*, 26, 3–24.

[35] Halman, N., Klabjan, D., Mostagir, M., Orlin, J., and Simchi-Levi, D. (2009), A Fully Polynomial-Time Approximation Scheme for Single-Item Stochastic Inventory Control with Discrete Demand, *Mathematics of Operations Research*, 34(3), 674–685.

[36] Guan, Y. and Miller, A. J. (2008), Polynomial-Time Algorithms for Stochastic Uncapacitated Lot-Sizing Problems, *Operations Research*, 56(5), 1172–1183.

[37] Huang, K. and Kucukyavuz, S. (2008), On Stochastic Lot-Sizing Problems with Random Lead Times, *Operations Research Letters*, 36, 303–308.

List of symbols/abbreviations/acronyms/initialisms

BinRecN	The Range-Approximated Algorithm for MaxRhoSTM
BinRecT	The Ratio-Approximated Algorithm for MaxRhoSTM
CAF	Canadian Armed Forces
GAO U.S.	General Accounting Office of the United States
HPP	Homogeneous Poison process
IID	Independent and Identically Distributed
LORA	Level of Repair Analysis
LRU	Line Replaceable Unit
MaxRho	The Fleet-reliability-maximization Model
MaxRhoSTM	The STM Presentation of MaxRho
MDP	Markov Decision Process
MinCost	The Operating-cost-minimization Model
MinCostSTM	The STM Presentation of MinCost
MSO	Multi-stage Stochastic Optimization
PE	Prime Equipment
Recursion	The Algorithm for MinCostSTM
SDP	Stochastic Dynamic Programming
STB	Scenario Tree Based
SPS	Spare Parts Stocking
STM	Scenario Tree Model
US DoD	United States Department of Defense

Annex A: Notations I

T	There are T mutually-disjointed time periods $[t, t+1)$, where $t = 1, 2, \cdots, T$.
E_n	At time one, there are N failed PEs, denoted by $E_1, E_2, \cdots, E_N$.
L_m	Each PE is made up of M different LRUs, denoted by $L_1, L_2, \cdots, L_M$.
s_m	The number of spare L_m, which are *individually existed* in the system.
f_m	The number of functioning L_m at time one.
g_m	The number of malfunctioning L_m at time one.
q_m	The number of L_m due in (under repair or in transshipment) at time one.
n^t	The number of failed PEs (or PE frames) at time t, where $t = 1, 2, \cdots, T$.
$\hat{n}^t$	The number of recovered PEs during time period t, where $t = 1, 2, \cdots, T$.
$\mathbf{f}^t$	The number of functioning LRUs at time t, where $\mathbf{f}^t = (f_1^t, f_2^t, \cdots, f_M^t)$.
$\mathbf{g}^t$	The number of malfunctioning LRUs at time t, where $\mathbf{g}^t = (g_1^t, g_2^t, \cdots, g_M^t)$.
$\mathbf{q}^t$	The number of due-in LRUs at time t, where $\mathbf{f}^t = (q_1^t, q_2^t, \cdots, q_M^t)$.
k_m	The number of failed L_m of a PE failure with $k = \sum_{m=1}^{M} k_m$ failed PEs.
$\mathbf{k}$	The M-entry vector to denote a PE failure, i.e., $\mathbf{k} = (k_1, k_2, \cdots, k_M)$.
λ	It takes in average $\frac{1}{\lambda}$ time units for a functioning PE to get failed.
β_m	The probability of a failed PE due to a failed L_m, where $1 = \sum_{m=1}^{M} \beta_m$.
γ	The threshold probability to specify zero-probability PE-failure events.
K	The largest number of failed PEs defined by γ such as $\frac{\gamma^K e^{-\lambda}}{K!} \geq \gamma$.
x_m^t	The decision variable to denote L_m-repair operations at time t.
$\mathbf{x}^t$	The vector to denote repair operations at time t, i.e., $\mathbf{x}^t = (x_1^t, x_2^t, \cdots, x_M^t)$.
d_m^t	The lead time for sending a failed L_m to the depot for repair at time t.
α	The interests gained on per un-used fund per time period.
r_m	The variable cost for each L_m repair operation, which is time-independent.
h^t	The fixed cost for transporting one or more LRUs to the depot at time t.
c^t	The total operating cost of period t, where $t = 1, 2, \cdots, T$.
ρ^t	The recovering ratio of the first t periods, where $t = 1, 2, \cdots, T$.
A	The required/designated level of fleet reliability in MinCost.
B	The funds for repair and transportation operations in MaxRho.

This page intentionally left blank.

Annex B: Notations II

$\mathcal{T}$	The notation for a scenario tree with a default root node 1.
$\mathcal{V}$	The set of all nodes on $\mathcal{T}$, where index 1 is reserved for root node.
Γ	The number of PE failures or edges branched from a non-leaf node.
$\mathcal{T}(j)$	The subtree rooted on a non-leaf node j. In particular, $\mathcal{T}(1) = \mathcal{T}$.
$\mathcal{V}(j)$	The set of all nodes on $\mathcal{T}(j)$. In particular, $\mathcal{V}(1) = \mathcal{V}$.
$l(j)$	The level of node j with $1 \leq l(j) \leq T+1$, $\forall j \in \mathcal{V}$.
$\mathcal{L}(l)$	The set of nodes with level l, i.e., $\mathcal{L}(l) = \{j \mid l(j) = l, j \in \mathcal{V}\}$.
$a(j)$	The direct ancestor of node j. In particular, $a(1) = \emptyset$.
$\mathcal{D}(j)$	The set of direct descendant nodes of any non-leaf node j.
$\mathcal{C}(j,l)$	The set of nodes, which are in $\mathcal{V}(j)$ with l levels higher than node j.
$\mathcal{P}(j)$	The path from node 1 to node j including nodes 1 and j.
p_j	The probability of node j. In particular, $p_1 = 1$.
$D_{(j,m)}$	The L_m demands on node j. In particular, $D_{(1,m)} = 0$ for all m.
$\mathbf{D}_j$	The demand vector on node j, where $\mathbf{D}_j = (D_{(j,1)}, D_{(j,2)}, \cdots, D_{(j,M)})$.
$\bar{D}_{(j,m)}$	The F-demand of L_m on node j. In particular, $\bar{D}_{(1,m)} = 0$ for all m.
$\bar{\mathbf{D}}_j$	The F-demand vector on node j with $\bar{\mathbf{D}}_j = (\bar{D}_{(j,1)}, \bar{D}_{(j,2)}, \cdots, \bar{D}_{(j,M)})$.
$\mathbf{k}_j$	The PE failure denoted by the edge heading into node j.
$k_{(j,m)}$	The number of failed L_m of the PE failure denoted by $\mathbf{k}_j$.
$x_{(j,m)}$	The decision variable denoting the L_m-repair operations on node j.
y_j	The decision variable denoting if there are any LRU repairs on node j.
$d_{(j,m)}$	The required lead time for L_m-repair operations on node j.
r_m	The variable cost for each L_m-repair operation on all nodes.
h_j	The fixed cost for transporting LRUs to the depot on node j.
$\mathcal{A}'(j,m)$	The set including all *mLeaf* nodes in $\mathcal{V}(j)$.
$\hat{\mathcal{B}}(j,m)$	The set including all *mGRoot* nodes in $\mathcal{V}(j)$.
$\bar{\mathcal{B}}(j,m)$	The set including all *mERoot* nodes in $\mathcal{V}(j)$.
$\mathcal{T}_m(j)$	The L_m-demand layer rooted on node j, e.g., $\mathcal{T}_m(1) = \mathcal{T}_m$.
$\mathcal{V}_m(j)$	The set of nodes on $\mathcal{T}_m(j)$, e.g., $\mathcal{V}_m(1) = \mathcal{V}_m$.
$\mathcal{T}_R(j)$	The residual tree of $\mathcal{T}(j)$, e.g., $\mathcal{T}_R(1) = \mathcal{T}_R$.
$\mathcal{V}_R(j)$	The set of nodes on $\mathcal{T}_R(j)$, e.g., $\mathcal{V}_R(1) = \mathcal{V}_R$.
ε_t	The approximation ratio, which can be used as stop criteria.
ε_n	The approximation range, which can be used as stop criteria.

This page intentionally left blank.

DOCUMENT CONTROL DATA

(Security markings for the title, abstract and indexing annotation must be entered when the document is Classified or Designated.)

1. ORIGINATOR (The name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g. Centre sponsoring a contractor's report, or tasking agency, are entered in section 8.) DRDC – Centre for Operational Research and Analysis Dept. of National Defence, MGen G. R. Pearkes Bldg., 101 Colonel By Drive, 6CBS, Ottawa ON K1A 0K2, Canada	2a. SECURITY MARKING (Overall security marking of the document, including supplemental markings if applicable.) UNCLASSIFIED 2b. CONTROLLED GOODS (NON-CONTROLLED GOODS) DMC A REVIEW: GCEC APRIL 2011

3. TITLE (The complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S, C or U) in parentheses after the title.)

Detailed maintenance planning for military systems with random lead times and cannibalization

4. AUTHORS (Last name, followed by initials – ranks, titles, etc. not to be used.)

Zhang, R.; Ghanmi, A.

5. DATE OF PUBLICATION (Month and year of publication of document.)	6a. NO. OF PAGES (Total containing information. Include Annexes, Appendices, etc.)	6b. NO. OF REFS (Total cited in document.)
December 2014	48	37

7. DESCRIPTIVE NOTES (The category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.)

Scientific Report

8. SPONSORING ACTIVITY (The name of the department project office or laboratory sponsoring the research and development – include address.)

DRDC – Centre for Operational Research and Analysis
Dept. of National Defence, MGen G. R. Pearkes Bldg., 101 Colonel By Drive, 6CBS, Ottawa ON K1A 0K2, Canada

9a. PROJECT OR GRANT NO. (If appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant.)	9b. CONTRACT NO. (If appropriate, the applicable number under which the document was written.)
10a. ORIGINATOR'S DOCUMENT NUMBER (The official document number by which the document is identified by the originating activity. This number must be unique to this document.) DRDC-RDDC-2014-R165	10b. OTHER DOCUMENT NO(s). (Any other numbers which may be assigned this document either by the originator or by the sponsor.)

11. DOCUMENT AVAILABILITY (Any limitations on further dissemination of the document, other than those imposed by security classification.)

(X) Unlimited distribution
(   ) Defence departments and defence contractors; further distribution only as approved
(   ) Defence departments and Canadian defence contractors; further distribution only as approved
(   ) Government departments and agencies; further distribution only as approved
(   ) Defence departments; further distribution only as approved
(   ) Other (please specify):

12. DOCUMENT ANNOUNCEMENT (Any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution (beyond the audience specified in (11)) is possible, a wider announcement audience may be selected.)

Unlimited

13. ABSTRACT (A brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual.)

Detailed maintenance planning under uncertainty is one of the most important topics in military research and practice. As one of the fastest ways to recover failed weapon systems, cannibalization operations are commonly applied by maintenance personnel. Due to additional complexities introduced by these operations, detailed maintenance decision making with cannibalization was rarely studied in the literature. This report proposed an analytic model for making repair decisions in a multi-stage uncertain environment at the operational level, where cannibalization operations are allowed and repair lead times are random. The study addresses the problem of maintenance planning for military systems with random lead times and independent failures. The objective of the problem is to maximize fleet reliabilities under operating costs constraints. A complementary problem that minimizes total operating costs under fleet reliabilities constraints was also examined. A polynomial algorithm was proposed to solve the minimization problem and determine optimal decision strategies. This algorithm could be used as a subroutine in a binary-search algorithm to solve the maximization problem. The obtained solutions were proved to be controllable in such a way that solutions with designated approximation ratios were achievable by running the algorithm in predictable run times.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (Technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus. e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus identified. If it is not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title.)

Approximation
Cannibalization
Dynamically constrained production
Dynamic programming
Multi-item production planning
Multi-stage problem
Independent failures
Random lead times
Reliability
Scenario tree model
Limited budget

DRDC | RDDC

SCIENCE, TECHNOLOGY AND KNOWLEDGE
FOR CANADA'S DEFENCE AND SECURITY

SCIENCE, TECHNOLOGIE ET SAVOIR
POUR LA DÉFENSE ET LA SÉCURITÉ DU CANADA

www.drdc-rddc.gc.ca